

國立臺灣大學電機資訊學院資訊工程研究所



博士論文

Department of Computer Science and Information Engineering

College of Electrical Engineering and Computer Science

National Taiwan University

Doctoral Dissertation

統計式語言模型 – 語音文件標記、檢索以及摘要

Statistical Language Modeling –

Spoken Document Indexing, Retrieval and Summarization

陳冠宇

Kuan-Yu Chen

指導教授：陳信希 博士、王新民 博士

Advisor: Hsin-Hsi Chen, Ph.D. and Hsin-Min Wang, Ph.D.

中華民國 104 年 4 月

April, 2015



DEDICATION



This thesis is dedicated to my family and girlfriend!



中文摘要



由於越來越多唾手可得多媒體文件，促成了語音文件理解(Understanding)與組織(Organization)在過去二十幾年來成為重要的研究議題。在各式各樣的相關研究中，語音文件標記(Indexing)、檢索(Retrieval)以及語音摘要(Summarization)被視為是這個領域中重要且基礎的研究題目。統計式語言模型(Statistical Language Modeling)一直是一個有趣且極富挑戰的研究領域，其主要被用於量化一段文字在自然語言中存在的可能性。過去許多研究致力於將語言模型運用於語音文件處理的任務之中，多數的研究呈現了豐富且卓越的實驗成果。有鑑於語言模型對於語音文件處理的重要性，本論文將以語言模型為主軸，繼續深究語音文件標記、檢索與摘要等問題。

由於使用者所給定的查詢通常非常簡短，這是資訊檢索系統面臨的一項重要考驗，本論文從此問題出發，除了廣泛地研究前人所提出的方法外，並針對傳統的方法提出了一套統一化的見解，更將這項技術應用於語音文件摘要的問題之中；接著，受到 I-vector 技術的啟發，本論文提出一個新穎的語言模型方法，並進一步的與虛擬關聯回饋技術相結合，提升語音文件檢索的效能；我們也觀察到，雖然語言模型已被使用於語音文件摘要任務之中，但過去所用的技術皆是以單連語模型為主，無法考慮長距離的語意資訊，有鑑於此，本論文提出以遞迴式神經網路語言模型搭配課程學習法的訓練方式，成功地提升了語音文件摘要的成效；最後，語言模型的發展漸漸地由模型化轉變到向量化，本論文提出新穎的相似度評估方式，成功地與近年來陸續提出的各式詞向量表示法相匹配，運用於語音文件摘要的問題上，除此之外，本論文亦提出了機率式詞向量表示法，不僅繼承了傳統表示法的優點，更可以有效地彌補現今詞向量表示法詮釋性的不足。



ABSTRACT

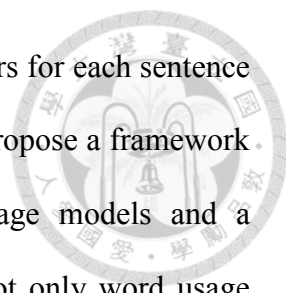


The inestimable volumes of multimedia associated with spoken documents that have been made available to the public in the past two decades have brought spoken document understanding and organization to the forefront as subjects of research. Among all the related subtasks, spoken document indexing, retrieval and summarization can be thought of as the cornerstones of this research area. Statistical language modeling (LM), which purports to quantify the acceptability of a given piece of text, has long been an interesting yet challenging research area. Much research shows that language modeling for spoken document processing has enjoyed remarkable empirical success. Motivated by the great importance of and interest in language modeling for various spoken document processing tasks (i.e., indexing, retrieval and summarization), language modeling is the backbone of this thesis.

In real-world applications, a serious challenge faced by the search engine is that queries usually consist of only a few words to address users' information needs. This thesis starts with a general survey of the practical challenge, and then not only proposes a principled framework which can unify the relationships among several widely-used approaches but also extends this school of techniques to spoken document summarization tasks.

Next, inspired by the concept of the i-vector technique, an i-vector based language modeling framework is proposed for spoken document retrieval and reformulated to accurately represent users' information needs.

Following, we are aware that language models have shown preliminary success in extractive speech summarization, but a central challenge facing the LM approach is how



to formulate sentence models and accurately estimate their parameters for each sentence in the spoken document to be summarized. Thus, in this thesis we propose a framework which builds on the notion of recurrent neural network language models and a curriculum learning strategy, which shows promise in capturing not only word usage cues but also long-span structural information about word co-occurrence relationships within spoken documents, thus eliminating the need for the strict bag-of-words assumption made by most existing LM-based methods.

Lastly, word embedding has been a recent popular research area due to its excellent performance in many natural language processing (NLP)-related tasks. However, as far as we are aware, there are relatively few studies that investigate its use in extractive text or speech summarization. First of all, this thesis focuses on building novel and efficient ranking models based on general word embedding methods for extractive speech summarization. Next, the thesis proposes a novel probabilistic modeling framework for learning word and sentence representations, which not only inherits the advantages of the original word embedding methods but also boasts a clear and rigorous probabilistic foundation.



CONTENTS



中文摘要	i
ABSTRACT	iii
CONTENTS	vi
LIST OF FIGURES	x
LIST OF TABLES	xii
Chapter 1 Introduction	1
1.1 Spoken Document Processing	1
1.2 Organization of the Thesis	3
Chapter 2 Overview of Related Literature	7
2.1 Statistical Language Modeling	7
2.1.1 Word-Regularity Language Modeling	7
2.1.2 Topic Language Modeling	10
2.1.3 Continuous Language Modeling	12
2.1.4 Neural Network-based Language Modeling	13
2.2 Spoken Document Indexing and Retrieval	15
2.2.1 Language Modeling for Spoken Document Retrieval	16
2.2.1.1 Query-Likelihood Measure	16
2.2.1.2 Kullback-Leibler (KL)-Divergence Measure	17
2.3 Speech Summarization	18
2.3.1 Language Modeling for Speech Summarization	20
Chapter 3 Speech and Language Corpora & Evaluation Metrics	24
3.1 Data Sets for Spoken Document Indexing and Retrieval	24
3.1.1 Subword-level Index Units	25

3.1.2	Evaluation Metrics	26
3.1.3	Baseline Experiments	27
3.2	Data Sets for Speech Summarization	28
3.2.1	Performance Evaluation	29
3.2.2	Baseline Experiments	31
Chapter 4	A Unified Framework for Pseudo-Relevance Feedback.....	34
4.1	Pseudo-Relevance Feedback	34
4.1.1	Relevance Modeling (RM).....	37
4.1.2	Simple Mixture Model (SMM)	37
4.1.3	Regularized Simple Mixture Model (RSMM)	38
4.2	A Unified Framework	39
4.3	Query-specific Mixture Modeling (QMM)	41
4.4	Experimental Results	42
Chapter 5	An I-vector based Language Modeling Framework for Retrieval	46
5.1	I-vector based Language Modeling	46
5.1.1	Experimental Results	50
5.2	Improved Query Representation with IVLM	52
5.2.1	Sample Pooling	53
5.2.2	I-vector Pooling.....	54
5.2.3	Model Pooling.....	54
5.2.4	Experimental Results	55
Chapter 6	A RNNLM-based Framework for Summarization.....	59
6.1	Recurrent Neural Network Language Modeling for Speech Summarization	59
6.2	Experimental Results	65
6.2.1	Experiments on Higher-order N -gram and Topic Language Modeling	65



6.2.2	Experiments on the Proposed RNNLM Summarizer	67
6.2.3	More Empirical Analysis of the RNNLM Summarizer	70
6.2.4	Further Extensions on RNNLM Summarizer	71
6.2.5	RNNLM with Syllable-level Index Units	73
6.2.6	Coupling RNNLM with Extra Acoustic Features	74
Chapter 7	A Word Embedding Framework for Summarization.....	78
7.1	Classic Word Embedding Methods.....	78
7.1.1	Continuous Bag-of-Words (CBOW) Model	78
7.1.2	Skip-Gram (SG) Model.....	79
7.1.3	Global Vector (GloVe) Model	80
7.1.4	Analytic Comparisons	81
7.2	Leveraging Word Embeddings for Summarization	82
7.2.1	Cosine Similarity Measure	82
7.2.2	The Triplet Learning Model	83
7.2.3	Document Likelihood Measure	85
7.2.4	Experimental Results	86
7.3	Probabilistic Word Embeddings	88
7.3.1	Probabilistic Bag-of-Words (PBOW) Model	89
7.3.2	Probabilistic Skip-gram (PSG) Model	90
7.3.3	Analytic Comparisons	91
7.3.4	Experimental Results	92
Chapter 8	Conclusion and Outlook.....	96
	REFERENCE	101



LIST OF FIGURES



Figure 2.1 Several state-of-the-art language models.....	8
Figure 4.1 A toy example of a user goes to a search engine.....	35
Figure 4.2 A schematic illustration of the SDR process with pseudo-relevance feedback.....	36
Figure 5.1 Retrieval results (in MAP) of i-vector based query representation techniques, relevance model (RM), and simple mixture model (SMM) with word- and subword-level index features.....	56
Figure 6.1 A schematic depiction of the fundamental network of RNNLM.	60
Figure 6.2 A sketch of the proposed RNNLM summarization framework.	62
Figure 6.3 Summarization results (in ROUGE-2) for each individual document (represented with either manual or speech transcript) in the test set, respectively, achieved by ULM and RNNLM+ULM.....	68
Figure 7.1 A running toy example for learning disparate distributional representations of a specific word w_5 , where the training corpus contains three documents, the vocabulary size is 9 (i.e., having words w_1, \dots, w_9) and the context window size is 1 (i.e., $c=1$).....	90
Figure 8.1 The important language models and the proposed frameworks are summarized year by year.	97



LIST OF TABLES



Table 3.1	Statistics for TDT-2 collection used for spoken document retrieval.....	25
Table 3.2	Retrieval results (in MAP) of different retrieval models with word- and subword-level index features.....	27
Table 3.3	The statistical information of the broadcast news documents used for the summarization.....	29
Table 3.4	The agreement among the subjects for important sentence ranking for the evaluation set.	30
Table 3.5	Summarization results achieved by a few well-studied or/and state-of-the-art unsupervised methods.....	31
Table 4.1	The summarization results (in F-scores(%)) achieved by various language models along with text and spoken documents.	43
Table 5.1	Retrieval results (in MAP) of IVLM with word- and subword-level index features for short and long queries using inductive and transductive learning strategies.	50
Table 5.2	Retrieval results (in MAP) of different approaches with word- and subword-level index features for short and long queries.....	51
Table 5.3	Retrieval results (in MAP) of different pooling methods with word- and subword-level index features with respect to the number of references ($ \mathbf{R} $).55	
Table 6.1	Training of RNNLM-based sentence models and the application of them for important sentence ranking.....	63
Table 6.2	Summarization results achieved by various LM-based methods, including ULM, BLM, PLSA, PLSA+ULM, RNNLM and RNNLM+ULM.	66

Table 6.3 Summarization results respectively achieved by ULM and RNNLM+ULM with respect to different summarization ratios.	67.
Table 6.4 Summarization results achieved by the proposed framework and a few well-studied or/and state-of-the-art unsupervised methods, which were measured by using the abstractive summaries written by the human subjects as the ground truth.	71
Table 6.5 Summarization results achieved by RNNLM+ULM with respect to different numbers of hidden-layer neurons being used.	72
Table 6.6 Summarization results achieved by RNNLM+ULM, MMR, ILP and their combinations.	74
Table 6.7 Summarization results achieved by ULM, RNNLM and RNNLM+ULM in conjunction with syllable-level index features.	74
Table 6.8 Four types of acoustic features used to represent each spoken sentence.	75
Table 6.9 Summarization results achieved by using acoustic features in isolation and its combination with ULM, RNNLM and ULM+RNNLM based sentence ranking scores, respectively.	75
Table 7.1 Summarization results achieved by various word-embedding methods in conjunction with the cosine similarity measure.	87
Table 7.2 Summarization results achieved by various word-embedding methods in conjunction with the triplet learning model.	87
Table 7.3 Summarization results achieved by various word-embedding methods in conjunction with the document likelihood measure.	87
Table 7.4 Summarization results achieved by various word-embedding methods in conjunction with the cosine similarity measure.	93
Table 7.5 Summarization results achieved by various word-embedding methods in	

conjunction with the document likelihood measure.93





Chapter 1 Introduction

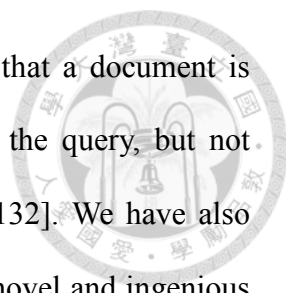


Before 2000, speech and text processing were two representative research areas, individually. Speech recognition [79][82], speaker identification and verification [168], voice synthesis [3] and so forth were important subtopics in the speech processing community. At the same time, information retrieval [177][179], language modeling [82][165], and summarization [130][131] were popular directions for text processing research. Since then, the rapid development of technology (especially computing hardware), the popularity of the Internet, and the rise of handheld devices have led to a considerable amount of research in spoken document processing [22][100][106].

1.1 Spoken Document Processing

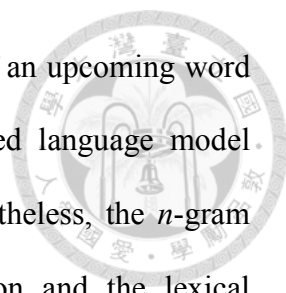
Along with the growing popularity of Internet applications, ever-increasing volumes of multimedia, such as broadcast radio and television programs, lecture recordings, and digital archives, are being made available in our daily life. Clearly, speech itself is one of the most important sources of information within multimedia. Users can efficiently listen to and digest multimedia associated with spoken documents by virtue of spoken content processing, which includes spoken document indexing, retrieval, and summarization [60][109][158].

A significant amount of effort has been put towards researching robust indexing (or representation) techniques [51][77][150] so as to extract probable spoken terms or phrases inherent in a spoken document that can match query words or phrases literally. On the other hand, spoken document retrieval (SDR), which revolves more around the notion of the relevance of a spoken document in response to a query, has also been a



prominent subject of much recent research. It is generally agreed that a document is relevant to a query if it addresses the stated information need of the query, but not merely because it happens to contain all the words in the query [132]. We have also witnessed a flurry of research activity aimed at the development of novel and ingenious methods for speech summarization, the aim of which is to generate a concise summary to help users efficiently review or quickly assimilate the important information conveyed by either a single spoken document or multiple spoken documents [59][125][136][151][163][206]. The dramatic growth of these studies is due in large part to advances in automatic speech recognition [60][157] and the ever-increasing volumes of multimedia associated with spoken documents made available to the public [109][158].

Beginning in the late 20th century, statistical language modeling has been successfully applied to various NLP-related applications, such as speech recognition [37][79][83], information retrieval [165][186][202], document summarization [25][29][115], and spelling error detection and correction [33][40][118][197]. Language modeling (LM) provides a statistical mechanism to associate quantitative scores to sequences of words or tokens. By far, the most widely-used and well-practiced language model is the n -gram language model [37][82], because of its simplicity and moderately good predictive power. For instance, in speech recognition, it can be used to constrain the acoustic analysis, guide the search through the vast space of candidate word strings, and quantify the acceptability of the final output from the speech recognizer [156][200]. This statistical paradigm was first introduced for information retrieval (IR) problems by Ponte and Croft (1998) [165], Song and Croft (1999) [186] and Miller, Leek, and Schwartz (1999) [143], demonstrating good success, and was then extended in a number of publications [31][50][99][203].



The n -gram language model, which determines the probability of an upcoming word given the previous $n-1$ word history, is the most commonly used language model because of its neat formulation and good predictive power. Nevertheless, the n -gram language model, as it only captures local contextual information and the lexical regularity of a language, is inevitably faced with two fundamental problems. On one hand, it is brittle across domains, since its performance is sensitive to changes in the genre or topic of the text on which it is trained. On the other hand, due to its limitation in scope, it fails to capture information (either semantic or syntactic) conveyed in the contextual history beyond its order (e.g., a trigram language model is limited to two words of context).

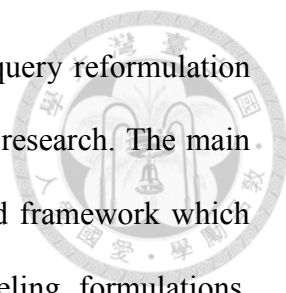
Motivated by the great importance of and interest in language modeling for various spoken document processing tasks, language modeling is the backbone of this thesis. Three subtasks (spoken document indexing, retrieval and summarization) are considered, and several insights are shared and methods proposed to unify conventional approaches or make further progress in complementing spoken document processing.

1.2 Organization of the Thesis

The remainder of this thesis is organized as follows:

Chapter 2 is a brief introduction to statistical language modeling, including word-regularity models, topic models, continuous language models and neural network-based language models. Also, spoken document indexing, retrieval and summarization are discussed.

Chapter 3 presents the experimental data sets, settings, and evaluation metrics for spoken document retrieval and summarization, as well as the baseline results.



Chapter 4 focuses on analyzing pseudo-relevance feedback for query reformulation approaches, and then presents a continuation of this general line of research. The main contribution here is two-fold. First, the thesis proposes a principled framework which unifies the relationships among several widely-used query modeling formulations. Second, on top of this successfully developed framework, an extended query modeling formulation is introduced by incorporating critical query-specific information cues to guide model estimation.

In Chapter 5 an i-vector based language modeling framework, stemming from the state-of-the-art i-vector framework for language identification and speaker recognition, is proposed and formulated to represent documents for spoken document retrieval. Also described in detail in this chapter are three novel methods to be applied in concert with i-vector based language modeling to more accurately represent user information needs.

Chapter 6 proposes a novel and effective recurrent neural network language modeling framework for speech summarization, on top of which the deduced sentence models are able to render not only word usage cues but also long-span structural information about word co-occurrence relationships within spoken documents, thus eliminating the need for the strict bag-of-words assumption. Second, the utility of the method originated from the proposed framework and that of several widely-used unsupervised methods are analyzed and compared extensively.

Beyond the effort made to improve word representations, Chapter 7 focuses on building novel and efficient ranking models based on general word embedding methods for extractive speech summarization. After that, the chapter also introduces a novel probabilistic modeling framework for learning word and sentence representations, which not only inherits the advantages from the original word embedding methods but also boasts a clear and rigorous probabilistic foundation.

Finally, Chapter 8 summarizes the contribution of this thesis and concludes the work.





Chapter 2 Overview of Related Literature



2.1 Statistical Language Modeling

Language modeling is an important component in most natural language processing (NLP)-related tasks today. The wide array of language modeling methods that have been developed so far fall roughly into four main categories: 1) word-regularity language models, 2) topic language models, 3) continuous language models, and 4) neural network language models. In this chapter, we briefly review several well-known or state-of-the-art language models. Figure 2.1 highlights some but not all of the state-of-the-art and widely-used language models year by year.

2.1.1 Word-Regularity Language Modeling

Beginning in the late 20th century, statistical language modeling has been successfully applied to various NLP applications, such as speech recognition [37][82], information retrieval [102][103][165], document summarization [25][115], and spelling correction [33][118][197]. The most widely-used and mature language model, by far, is the n -gram language model [37][82], because of its simplicity and fair predictive power. Quantifying the quality of a word string in a natural language is the most common task. Take the trigram model for example: when given a word string $W_1^L = w_1, w_2, \dots, w_L$, the probability of the word string is approximated by the following product of a series of conditional probabilities [82]:

$$\begin{aligned} P(W_1^L) &= P(w_1) \prod_{l=2}^L P(w_l | W_1^{l-1}) \\ &\approx P(w_1) P(w_2 | w_1) \prod_{l=3}^L P(w_l | w_{l-2}, w_{l-1}). \end{aligned} \tag{2.1}$$

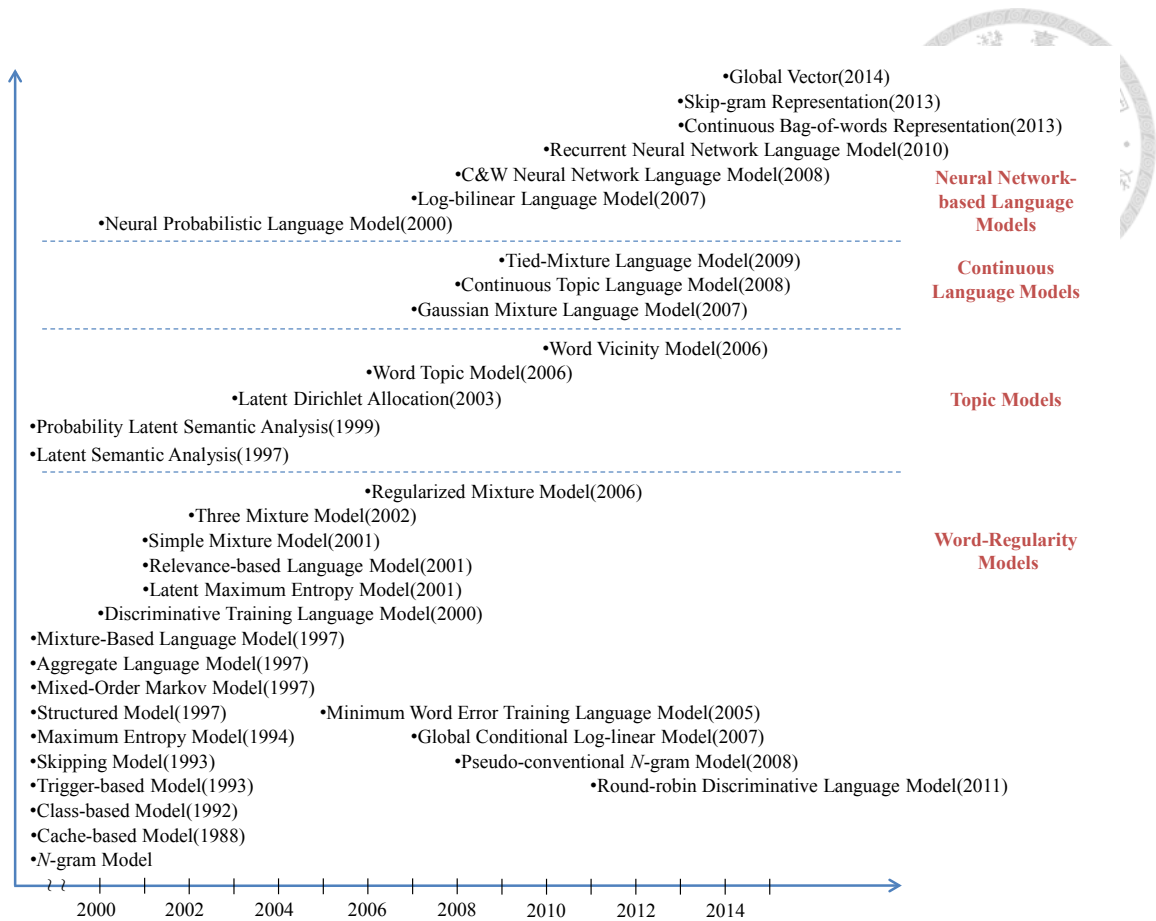


Figure 2.1 Several state-of-the-art language models.

In the trigram model, we make the approximation (or assumption) that the probability of a word depends only on the two immediately preceding words.

The easiest way to estimate the conditional probability in Eq. (2.1) is to use the maximum likelihood (ML) estimation

$$P(w_l | w_{l-2}, w_{l-1}) = \frac{c(w_{l-2}, w_{l-1}, w_l)}{c(w_{l-2}, w_{l-1})}, \quad (2.2)$$

where $c(w_{l-2}, w_{l-1}, w_l)$ and $c(w_{l-2}, w_{l-1})$ denote the occurrences of the word strings “ w_{l-2}, w_{l-1}, w_l ” and “ w_{l-2}, w_{l-1} ” in a given training corpus, respectively. Without loss of generality, the trigram model can be extended to higher order models, such as the four-gram model and the five-gram model, but the high-order n -gram models usually suffer from data sparseness, which leads to zero conditional probabilities. To eliminate

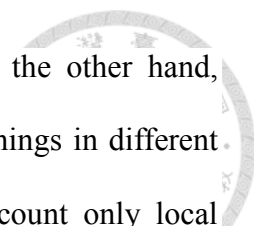
zero probabilities, various smoothing techniques have been proposed, e.g., Good-Turing [66][88], Kneser-Ney [37][92], and Pitman-Yor [78]. The general formulation of these approaches is [37]

$$P(w_l | w_{l-n+1}, \dots, w_{l-1}) = \begin{cases} f(c(w_{l-n+1}, \dots, w_l)) & , \text{ if } c(w_{l-n+1}, \dots, w_l) \neq 0 \\ \beta(w_{l-n+1}, \dots, w_{l-1})f(c(w_{l-n+1}, \dots, w_l)), & \text{ if } c(w_{l-n+1}, \dots, w_l) = 0 \end{cases} \quad (2.3)$$

where $f(\cdot)$ denotes a discounting probability function and $\beta(\cdot)$ denotes a back-off weighting factor that makes the distribution sum to one.

Clearly, n -gram language modeling focuses on modeling the local contextual information or the lexical regularity of a language, and it is recognized as the earliest language model. Continuing this school of research, many successive language models have been proposed, such as the cache language model [95][96], the trigger model [101], the class-based language models [17][196] and the maximum entropy language model [174][175]. Interested readers are referred [176] to for thorough and entertaining discussions on the major methods.

The year 1997 can be thought as a watershed in language modeling research. On one hand, discriminative language modeling [41][208] is representative of the following research. Although this sort of research still is aimed at building n -gram models, the major difference between discriminative language modeling and conventional n -gram models is the training objective. Conventional n -gram-based language models seek a set of parameters by maximizing the corpus likelihood with a ML criterion, while discriminative language models seek parameters that reduce the speech recognition error rate [41], enhance the F-score for information retrieval [23], or optimize the rouge score for summarization [120]. The minimum word error training (MERT) [153][155], the global conditional log-linear model (GCLM) [170][171], and the round-robin

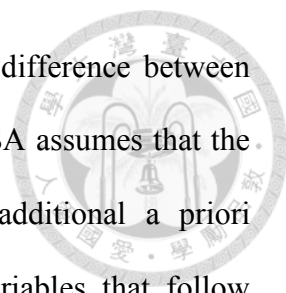


discriminative model (R2D2) [154] are representative methods. On the other hand, researchers are aware that many polysemic words have different meanings in different contexts, and that word-regularity language modeling takes into account only local context information, and thus cannot capture long-span semantic information embedded in a sentence or document. To mitigate this flaw, topic language modeling has been proposed [14][26][75]. We give a brief introduction to this school of research in the next subsection.

2.1.2 Topic Language Modeling

The n -gram language model, as it is aimed at capturing only local contextual information, or a language's lexical regularities, is unable to capture information (either semantic or syntactic) conveyed by words before the $n-1$ immediately preceding words. To mitigate this weakness of the n -gram model, various topic language models have been proposed and widely used in many NLP tasks. We can roughly organize these topic models into two categories [30][31]: document topic models (DTMs) and word topic models (WTMs).

DTMs introduce a set of latent topic variables to describe the “word-document” co-occurrence characteristics. The dependence between a word and its preceding words (regarded as a document) is not computed directly based on frequency counts as in the conventional n -gram model. The probability now is instead based on the frequency of the word in the latent topics as well as the likelihood that the preceding words together generate the respective topics. Probabilistic latent semantic analysis (PLSA) [75][76] and latent Dirichlet allocation (LDA) [12][67] are two representatives of this category. LDA, having a formula analogous to PLSA, can be regarded as an extension to PLSA



and has enjoyed much success for various NLP tasks. The major difference between PLSA and LDA is the inference of model parameters [12][14]. PLSA assumes that the model parameters are fixed and unknown while LDA places additional a priori constraints on model parameters by treating them as random variables that follow Dirichlet distributions. Since LDA has a more complex form for model optimization, which is not easily solved by exact inference, several approximate inference algorithms, including variational approximation [11][12][14], expectation propagation [145], and Gibbs sampling [67], have been proposed to estimate LDA parameters.

Instead of treating the preceding word string as a document topic model, we can further regard each word w_l of the language as a word topic model (WTM) [44][45]. Each WTM model M_{w_l} can be trained in a data-driven manner by concatenating those words occurring within the vicinity of each occurrence of w_l in a training corpus, which are postulated to be relevant to w_l . To this end, a sliding window with a size of S words is placed on each occurrence of w_l , allowing for the consequent aggregation of a pseudo-document associated with such vicinity information of w_l . The WTM model of each word can be estimated using the expectation-maximization (EM) algorithm [52] by maximizing the total log-likelihood of words occurring in their associated “vicinity documents”. The word vicinity model (WVM) [30] bears a certain similarity to WTM in its motivation of modeling word-word co-occurrences, but has a more concise parameterization. WVM explores word vicinity information by directly modeling the joint probability of any word pair in the language. Along a similar vein, WVM is trained using the EM algorithm by maximizing the probabilities of all word pairs, respectively, that co-occur within a sliding window of S words in the training corpus.

It is worth noting that several variations of topic language models have been

proposed for use with NLP-related tasks, including the supervised topic model [13], the labeled LDA [167], and the latent association analysis (LAA) [137].



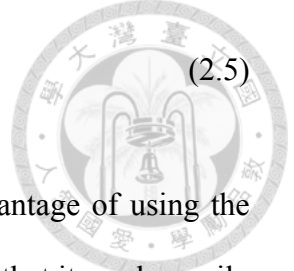
2.1.3 Continuous Language Modeling

The fundamental theorem of the Gaussian mixture language model (GMLM) was proposed in 2007 [1]. The GMLM model claims that although the n -gram has been the dominant and successful technology for language modeling, its two greatest weaknesses are clearly: *generalizability* and *adaptability*. To leverage the lessons learned in acoustic modeling for speech recognition, GMLM is an attempt to use Gaussian mixture models to model language instead of the usual multinomial distributions. Formally, GMLM employs singular value decomposition (SVD) [51] to project each word in the vocabulary to a continuous space, thus assigning to each word its own distributed representation. Since each history consists of a set of words of size $n-1$ for an n -gram sample, the history can be represented by concatenating the word representations corresponding to the words in the history. GMLM then models contextual information by using a Gaussian mixture model (GMM) [11] for each word respectively. Specifically, word w_i has its own density function with which to calculate the probability densities for an observed history $W_{i-n+1}^{i-1} = w_{i-n+1}, \dots, w_{i-1}$:

$$p(W_{i-n+1}^{i-1} | w_i) = \sum_{m=1}^M c_{w_i, m} N(W_{i-n+1}^{i-1} | \mu_{w_i, m}, \Sigma_{w_i, m}) \quad (2.4)$$

where M is the number of mixtures in the GMM model of word w_i , and $c_{w_i, m}$, $\mu_{w_i, m}$, and $\Sigma_{w_i, m}$ are the component weight, mean vector, and covariance matrix for the m -th mixture in the GMM model. However, when we observe a history, what we need is the prediction probability. GMLM suggests using Bayes' rule to calculate the conditional probability as

$$P(w_i | W_{i-n+1}^{i-1}) = \frac{P(w_i)p(W_{i-n+1}^{i-1} | w_i)}{p(W_{i-n+1}^{i-1})} = \frac{P(w_i)p(W_{i-n+1}^{i-1} | w_i)}{\sum_{w_j \in V} P(w_j)p(W_{i-n+1}^{i-1} | w_j)} \quad (2.5)$$



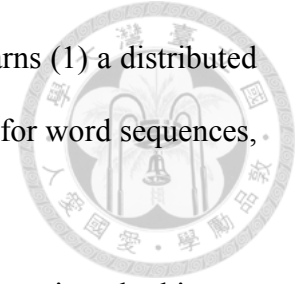
where $P(w_i)$ is the conventional unigram language model. The advantage of using the Gaussian mixture model as the cornerstone of the language model is that it can be easily adapted from a relatively small size of text corpus by utilizing well-studied techniques such as maximum likelihood linear regression (MLLR) [110]. Moreover, it is also well-suited for combination with the concept of clusters to enhance generalization capability. Continuing this school of research, many related language models have been proposed, such as the tied-mixture language modeling (TMLM) [180] and the continuous topic language modeling (CTLM) [46].

2.1.4 Neural Network-based Language Modeling

The artificial neural network (ANN) can be dated back to the threshold logic which is a computational model for neural networks based on mathematics and algorithms [134]. Although several neural network-based language models have been proposed year by year, this school of research gained attention only after the year 2000. Feedforward neural network [68] and recurrent neural network [56] are two important representatives.

The feedforward neural network language model (NNLM) is n -gram-based language modeling [7][8]. The original motivation for NNLM was to mitigate the data scantiness faced by conventional n -gram models. A famous example is the sentence “*The cat is walking in the bedroom*”: seeing this sentence in the training corpus, we should generalize such that the sentence “*A dog was running in a room*” is almost as likely, simply because *dog* and *cat* (or *the* and *a*, *room* and *bedroom*, and so on) have similar

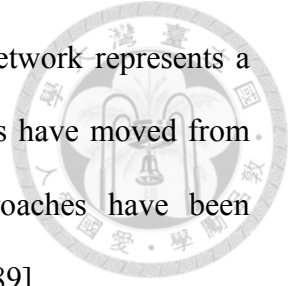
semantic and grammatical roles. To achieve this goal, the model learns (1) a distributed representation for each word along with (2) the probability function for word sequences, expressed in terms of these representations, simultaneously.



The recurrent neural network language model (RNNLM) tries to project the history, W_1^{L-1} , onto a continuous space and estimate the conditional probability in a recursive way by using the full information about W_1^{L-1} [36][138]. It has recently emerged as a promising language modeling framework that can effectively and efficiently capture the long-span context relationships among words (or more precisely, the dependence between an upcoming word and its whole history) for use in speech recognition and spoken document summarization. The fundamental network of RNNLM consists of three main ingredients: the input layer, the hidden layer and the output layer. The most attractive aspect of RNNLM is that the statistical cues of previously encountered words retained in the hidden layer are fed back to the input layer and work in combination with the currently encountered word w_{L-1} as an “augmented” input vector for predicting an arbitrary succeeding word w_L . Intuitively, the information stored in the hidden layer can be viewed as topic-like information similar in spirit to PLSA or LDA; the major difference is that RNNLM leverages a set of non-linear active functions to calculate the values of the latent variables while PLSA or LDA estimates the corresponding model parameters by using the (variational) EM algorithm. Thus doing, RNNLM naturally takes into account not only word usage cues but also long-span structural information about word co-occurrence relationships for language modeling.

Recently, neural networks have emerged as a popular subject of research because of their excellent performance in many fundamental areas, including multimedia processing [87][94], speech processing [1][148], and natural language processing

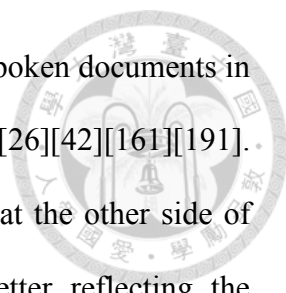
[48][146][183]. In language model research, the recurrent neural network represents a breakthrough in building language models; recently, research trends have moved from modeling to vectorization. Several representation learning approaches have been proposed and applied to various NLP-related tasks [10][124][159][189].



2.2 Spoken Document Indexing and Retrieval

Over the last two decades, spoken document retrieval (SDR) has become an active area of research and experimentation in the speech processing community. Although most retrieval systems participating in the TREC-SDR evaluations had claimed that speech recognition errors do not seem to cause much adverse effect on SDR performance when merely using imperfect recognition transcripts derived from one-best recognition results from a speech recognizer, this is probably due to the fact that the TREC-style test queries tend to be quite long and contain different words describing similar concepts that could help the queries match their relevant spoken documents. Furthermore, a query word (or phrase) might occur repeatedly (more than once) within a relevant spoken document, and it is not always the case that all of the occurrences of the word would be misrecognized totally as other words. Nevertheless, we believe that there are still at least two fundamental challenges facing SDR. On one hand, the imperfect speech recognition transcript carries wrong information and thus would deviate somewhat from representing the true theme of a spoken document. On the other hand, a query is often only a vague expression of an underlying information need, and there probably would be word usage mismatch between a query and a spoken document even if they are topically related to each other.

A significant body of spoken content retrieval work has been placed on the



exploration of robust indexing or modeling techniques to represent spoken documents in order to work around (or mitigate) the problems caused by ASR [22][26][42][161][191]. On the contrary, very limited research has been conducted to look at the other side of the coin, namely, the improvement of query formulation for better reflecting the underlying information need of a user [27]. As for the latter problem, pseudo-relevance feedback [1][173] is by far the most commonly-used paradigm, which assumes that a small amount of top-ranked spoken documents obtained from the initial round of retrieval are relevant and can be utilized for query reformulation. Subsequently, the retrieval system can perform a second round of retrieval with the enhanced query representation to search for more relevant documents.

2.2.1 Language Modeling for Spoken Document Retrieval

2.2.1.1 Query-Likelihood Measure

Recently, language modeling (LM) has emerged as a promising approach to building SDR systems [26][27][42]. This is due to the fact that the LM approach has inherent clear probabilistic foundation and excellent retrieval performance [204]. The fundamental formulation of the LM approach to SDR is to compute the conditional probability $P(Q|D)$, i.e., the likelihood of a query Q generated by each spoken document D (the so-called query-likelihood measure). A spoken document D is deemed to be relevant with respect to the query Q if the corresponding document model is more likely to generate the query. If the query Q is treated as a sequence of words, $Q=w_1, w_2, \dots, w_L$, where the query words are assumed to be conditionally independent given the document D and their order is also assumed to be of no importance (i.e., the so-called “*bag-of-words*” assumption), the similarity measure $P(Q|D)$ can be further decomposed

as a product of the probabilities of the query words generated by the document [204]:

$$P(Q|D) = \prod_{l=1}^L P(w_l|D), \quad (2.6)$$

where $P(w_l|D)$ is the likelihood of generating w_l by document D (a.k.a. the document model). The simplest way to construct $P(w_l|D)$ is based on literal term matching [109], or using the unigram language model (ULM). To this end, each document D can, respectively, offer a unigram distribution for observing any given word w , which is parameterized on the basis of the empirical counts of words occurring in the document with the maximum likelihood (ML) estimator [82][204]:

$$P(w|D) = \frac{c(w,D)}{|D|}, \quad (2.7)$$

where $c(w,D)$ is the number of times that word w occurs in the document D and $|D|$ is the number of words in the document. The document model is further smoothed by a background unigram language model estimated from a large general collection to model the general properties of the language as well as to avoid the problem of zero probability [204]. However, how to strike the balance between these two probability distributions is actually a matter of judgment, or trial and error.

2.2.1.2 Kullback-Leibler (KL)-Divergence Measure

Another basic formulation of LM for SDR is the Kullback-Leibler (KL)-divergence measure [97][204]:

$$\begin{aligned} -KL(Q||D) &= - \sum_{w \in V} P(w|Q) \log \frac{P(w|Q)}{P(w|D)} \\ &\stackrel{\text{rank}}{=} \sum_{w \in V} P(w|Q) \log P(w|D), \end{aligned} \quad (2.8)$$

where the query and the document are, respectively, framed as a (unigram) language model (i.e., $P(w|Q)$ and $P(w|D)$), $\stackrel{\text{rank}}{=}$ means equivalent in terms of being used for the

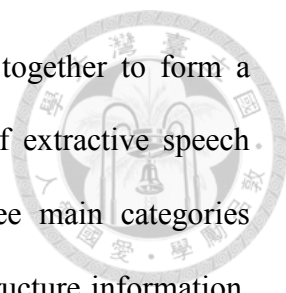
purpose of ranking documents, and V denotes the vocabulary. A document D has a smaller value (or probability distance) in terms of $KL(Q||D)$ is deemed to be more relevant with respect to Q . The retrieval effectiveness of the KL-divergence measure depends primarily on the accurate estimation of the query modeling $P(w|Q)$ and the document modeling $P(w|D)$. In addition, it is easy to show that the KL-divergence measure will give the same ranking as the ULM model (*cf.* Eq. (2.6) and Eq. (2.7)) when the query language model is simply derived with the ML estimator [27]:

$$\begin{aligned}
- KL(Q||D)^{\text{rank}} &= \sum_{w \in V} P(w|Q) \log P(w|D) \\
&= \sum_{w \in V} \frac{c(w, Q)}{|Q|} \log P(w|D) \\
&= \sum_{w \in V} c(w, Q) \log P(w|D) \\
&= \log P(Q|D) \\
&= P(Q|D).
\end{aligned} \tag{2.9}$$

In Eq. (2.9), $P(w|Q)$ is simply estimated as $c(w, Q)/|Q|$, where $c(w, Q)$ is the number of times w occurring in Q and $|Q|$ is the total count of words in Q . Accordingly, the KL-divergence measure not only can be thought as a generalization of the query-likelihood measure, but also has the additional merit of being able to accommodate extra information cues to improve the estimation of its component models (especially, the query model) for better document ranking in a systematic manner [27][204].

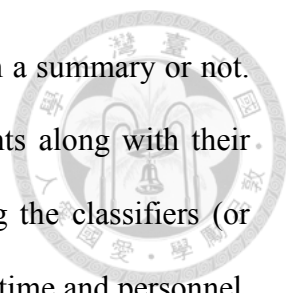
2.3 Speech Summarization

By virtue of extractive speech summarization, one can listen to and digest multimedia associated with spoken documents efficiently. Extractive speech summarization manages to select a set of indicative sentences from an original spoken document



according to a target summarization ratio and concatenates them together to form a summary accordingly [125][130][151][163]. The wide spectrum of extractive speech summarization methods developed so far may be split into three main categories [125][130]: 1) methods simply based on the sentence position or structure information, 2) methods based on unsupervised sentence ranking, and 3) methods based on supervised sentence classification.

For the first category, the important sentences can be selected from some salient parts of a spoken document [5]. For instance, sentences can be selected from the introductory and/or concluding parts of a spoken document. However, such methods can be only applied to some specific domains with limited document structures. On the other hand, unsupervised sentence ranking methods attempt to select important sentences based on statistical features of spoken sentences or of the words in the sentences without human labor involved. Statistical features, for example, can be the term (word) frequency, linguistic score and recognition confidence measure, as well as the prosodic information. The associated unsupervised methods based on these features have gained much attention of research. Among them, the vector space model (VSM) [65], the latent semantic analysis (LSA) method [65], the Markov random walk (MRW) method [192], the maximum marginal relevance (MMR) method [19], the sentence significant score method [59], the LexRank method [58], the submodularity-based method [114], and the integer linear programming (ILP) method [135] are the most popular approaches for spoken document summarization. Apart from that, a number of classification-based methods using various kinds of representative features also have been investigated, such as the Gaussian mixture models (GMM) [65], the Bayesian classifier (BC) [98], the support vector machine (SVM) [205] and the conditional random fields (CRFs) [61], to name just a few. In these methods, important sentence selection is usually formulated as



a binary classification problem. A sentence can either be included in a summary or not. These classification-based methods need a set of training documents along with their corresponding handcrafted summaries (or labeled data) for training the classifiers (or summarizers). However, manual annotation is expensive in terms of time and personnel. Even if the performance of unsupervised summarizers is not always comparable to that of supervised summarizers, their easy-to-implement and flexible property (i.e., they can be readily adapted and carried over to summarization tasks pertaining to different languages, genres or domains) still makes them attractive. Interested readers may also refer to [125][130][151][163] for comprehensive reviews and new insights into the major methods that have been developed and applied with good success to a wide range of text and speech summarization tasks.

2.3.1 Language Modeling for Speech Summarization

Among the aforementioned methods, one of the emerging lines of research is to employ the language modeling (LM) approach for important sentence selection, which has shown preliminary success for performing extractive speech summarization in an unsupervised fashion [38][116]. However, a central challenge facing the LM approach is how to formulate the sentence models and accurately estimate their parameters for each sentence in the spoken document to be summarized.

Intuitively, extractive speech summarization could be cast as an ad-hoc information retrieval (IR) problem, where a spoken document to be summarized is taken as an information need and each sentence of the document is regarded as a candidate information unit to be retrieved according to its relevance (or importance) to the information need. As such, the ultimate goal of extractive speech summarization could be stated as the selection of the most representative sentences that can succinctly

describe the main topics of the spoken document.

When applying the LM-based approach to extractive speech summarization, a principal realization is to use a probabilistic generative paradigm for ranking each sentence S of a spoken document D to be summarized, which can be expressed by $P(S|D)$. Instead of calculating this probability directly, we can apply the Bayes' rule and rewrite it as follows [82]:

$$P(S|D) = \frac{P(D|S)P(S)}{P(D)}, \quad (2.10)$$

where $P(D|S)$ is the sentence generative probability, i.e., the likelihood of D being generated by S , $P(S)$ is the prior probability of the sentence S being relevant, and $P(D)$ is the prior probability of the document D . $P(D)$ in Eq. (2.10) can be eliminated because it is identical for all sentences and will not affect the ranking of the sentences. Furthermore, because the way to estimate the probability $P(S)$ is still under active study [38], we may simply assume that $P(S)$ is uniformly distributed, or identical for all sentences. In this way, the sentences of a spoken document to be summarized can be ranked by means of the probability $P(D|S)$ instead of using the probability $P(S|D)$: the higher the probability $P(D|S)$, the more representative S is likely to be for D . If the document D is expressed as a sequence of words, $D=w_1, w_2, \dots, w_L$, where words are further assumed to be conditionally independent given the sentence and their order is assumed to be of no importance (i.e., the so-called “*bag-of-words*” assumption), then $P(D|S)$ can be approximated by

$$P(D|S) \approx \prod_{i=1}^L P(w_i|S), \quad (2.11)$$

where L denotes the length of the document D . The sentence ranking problem has now been reduced to the problem of how to accurately infer the probability distribution $P(w_i|S)$, i.e., the corresponding sentence model for each sentence of the document.

Again, the simplest way is to estimate a unigram language model (ULM) on the basis of the frequency of each distinct word w occurring in the sentence, with the maximum likelihood (ML) criterion [82][204]:

$$P(w|S) = \frac{c(w,S)}{|S|}, \quad (2.12)$$

where $c(w,S)$ is the number of times that word w occurs in S and $|S|$ is the length of S . The ULM model can be further smoothed by a background unigram language model estimated from a large general collection to model the general properties of the language as well as to avoid the problem of zero probability. It turns out that a sentence S with more document words w occurring frequently in it would tend to have a higher probability of generating the document.





Chapter 3 Speech and Language Corpora & Evaluation Metrics



3.1 Data Sets for Spoken Document Indexing and Retrieval

The thesis uses the Mandarin Chinese collection of the TDT corpora for the retrospective retrieval task [23][104], such that the statistics for the entire document collection is obtainable. The Chinese news stories (text) from Xinhua News Agency are used as our test queries and training corpus for all models (excluding test query set). More specifically, in the following experiments, we will merely extract the title field from a news story as a test query. The Mandarin news stories (audio) from Voice of America news broadcasts are used as the spoken documents. All news stories are exhaustively tagged with event-based topic labels, which serve as the relevance judgments for performance evaluation. Table 3.1 describes some basic statistics about the corpora used in this thesis. The Dragon large-vocabulary continuous speech recognizer provided Chinese word transcripts for our Mandarin audio collections (TDT-2). To assess the performance level of the recognizer, we spot-checked a fraction of the TDT-2 development set (about 39.90 hours) by comparing the Dragon recognition hypotheses with manual transcripts, and obtained a word error rate (WER) of 35.38%. Since Dragon's lexicon is not available, we augmented the LDC Mandarin Chinese Lexicon with 24k words extracted from Dragon's word recognition output, and for computing error rates used the augmented LDC lexicon (about 51,000 words) to tokenize the manual transcripts. We also used this augmented LDC lexicon to tokenize the query sets and training corpus in the retrieval experiments.

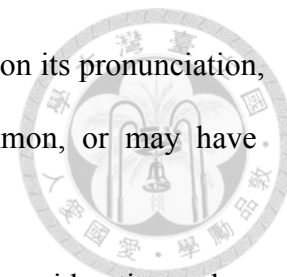
	TDT-2 (Development Set) 1998, 02~06			
# Spoken documents	2,265 stories, 46.03 hours of audio			
# Distinct test queries	16 Xinhua text stories (Topics 20001~20096)			
# Distinct training queries	819 Xinhua text stories (Topics 20001~20096)			
	Min.	Max.	Med.	Mean
Doc. length (in characters)	23	4,841	153	287.1
Short query length (in characters)	8	27	13	14
Long query length (in characters)	183	2,623	329	532.9
# Relevant documents per test query	2	95	13	29.3
# Relevant documents per training query	2	95	87	74.4

Table 3.1 Statistics for TDT-2 collection used for spoken document retrieval.

3.1.1 Subword-level Index Units

In Mandarin Chinese, there is an unknown number of words, although only some (e.g., 80 thousands, depending on the domain) are commonly used. Each word encompasses one or more characters, each of which is pronounced as a monosyllable and is a morpheme with its own meaning. Consequently, new words are easily generated every day by combining a few characters. Furthermore, Mandarin Chinese is phonologically compact; an inventory of about 400 base syllables provides full phonological coverage of Mandarin audio, if the differences in tones are disregarded. Additionally, an inventory of about 6,000 characters almost provides full textual coverage of written Chinese. There is a many-to-many mapping between characters and syllables. As such,

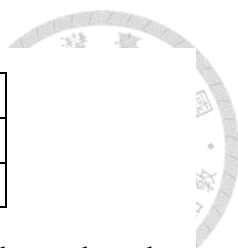
a foreign word can be translated into different Chinese words based on its pronunciation, where different translations usually have some syllables in common, or may have exactly the same syllables.



The characteristics of the Chinese language lead to some special considerations when performing Mandarin Chinese speech recognition; for example, syllable recognition is believed to be a key problem. Mandarin Chinese speech recognition evaluation is usually based on syllable and character accuracy, rather than word accuracy. The characteristics of the Chinese language also lead to some special considerations for SDR. Word-level indexing features possess more semantic information than subword-level features; hence, word-based retrieval enhances precision. On the other hand, subword-level indexing features behave more robustly against the Chinese word tokenization ambiguity, homophone ambiguity, open vocabulary problem, and speech recognition errors; hence, subword-based retrieval enhances recall. Accordingly, there is good reason to fuse the information obtained from indexing the features of different levels [23]. To do this, syllable pairs are taken as the basic units for indexing besides words. Both the manual transcript and the recognition transcript of each spoken document, in form of a word stream, were automatically converted into a stream of overlapping syllable pairs. Then, all the distinct syllable pairs occurring in the spoken document collection were identified to form a vocabulary of syllable pairs for indexing. We can simply use syllable pairs, in replace of words, to represent the spoken documents, and thereby construct the associated retrieval models.

3.1.2 Evaluation Metrics

The retrieval results are expressed in terms of non-interpolated mean average precision (MAP) following the TREC evaluation [63], which is computed by the following



	VSM	LSA	SCI	ULM	LDA
Word	0.273	0.296	0.270	0.321	0.328
Subword	0.257	0.384	0.270	0.329	0.377

Table 3.2 Retrieval results (in MAP) of different retrieval models with word- and subword-level index features.

equation:

$$\text{MAP} = \frac{1}{L} \sum_{i=1}^L \frac{1}{N_i} \sum_{j=1}^{N_i} \frac{j}{r_{i,j}}, \quad (3.1)$$

where L is the number of test queries, N_i is the total number of documents that are relevant to query Q_i , and $r_{i,j}$ is the position (rank) of the j -th document that is relevant to query Q_i , counting down from the top of the ranked list.

3.1.3 Baseline Experiments

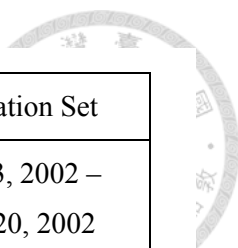
In the first set of experiments, we compare several retrieval models, including the vector space model (VSM) [132][178], the latent semantic analysis (LSA) [51], the semantic context inference (SCI) [77], and the basic LM-based method (i.e., ULM) [202]. The results when using word- and subword-level index features are shown in Table 3.2. At first glance, ULM in general outperforms the other three methods in most cases, validating the applicability of the LM framework for SDR. Next, we compare two extensions of ULM, namely the probabilistic latent semantic analysis (PLSA) [31] and the latent Dirichlet allocation (LDA) [195], with ULM. The experimental results are also shown in Table 3.2. As expected, both PLSA and LDA outperform ULM, and they are almost on par with each other. The results also reveal that PLSA and LDA can give more accurate estimates of the document language models than the empirical ML

estimator used in ULM, and thus improve the retrieval effectiveness. On the other hand, if we have a close look at these results, we notice that although the word error rate (WER) for the spoken document collection is higher than 35%, it does not lead to catastrophic failures probably due to the reason that recognition errors are overshadowed by a large number of spoken words correctly recognized in the documents.

3.2 Data Sets for Speech Summarization

The summarization dataset is a broadcast news corpus collected by the Academia Sinica and the Public Television Service Foundation of Taiwan between November 2001 and April 2003 [193], which has been segmented into separate stories and transcribed manually. Each story contains the speech of one studio anchor, as well as several field reporters and interviewees. A subset of 205 broadcast news documents compiled between November 2001 and August 2002 was reserved for the summarization experiments. Since broadcast news stories often follow a relatively regular structure as compared to other speech materials like conversations, the positional information would play an important role in extractive summarization of broadcast news stories. We hence chose 20 documents, for which the generation of reference summaries is less correlated with the positional information (or the position of sentences), as the held-out test set to evaluate the general performance of the proposed summarization framework, while another subset of 100 documents the held-out development set for tuning the parameters of the various unsupervised summarization methods compared in the thesis.

On the other hand, twenty-five hours of gender-balanced speech from the remaining speech data were used to train the acoustic models for speech recognition. The data was



	Training Set	Evaluation Set
Recording Period	Nov. 7, 2001 – Jan. 22, 2002	Jan. 23, 2002 – Aug. 20, 2002
Number of Documents	185	20
Average Duration per Document (in sec.)	129.4	141.28
Avg. Number of words per Document	326.0	290.3
Avg. Number of Sentences per Document	20.0	23.25
Avg. Word Error Rate (WER)	38.0%	39.4%

Table 3.3 The statistical information of the broadcast news documents used for the summarization.

first used to bootstrap the acoustic model training with the ML criterion. Then, the acoustic models were further optimized by the minimum phone error (MPE) discriminative training algorithm [73]. Table 3.3 shows some basic statistics about the spoken documents of the development and evaluation sets, where the average word error rate (WER) obtained for the spoken documents was about 38.1% [24]. A large number of text news documents collected by the Central News Agency (CNA) between 1991 and 2002 (the Chinese Gigaword Corpus released by LDC) were used. The documents collected in 2000 and 2001 were used to train N -gram language models for speech recognition with the SRI Language Modeling Toolkit [187].

3.2.1 Performance Evaluation

Three subjects were asked to create extractive summaries of the 205 spoken documents for the summarization experiments as references (the gold standard) for evaluation. The

Kappa	ROUGE-1	ROUGE-2	ROUGE-L
0.544	0.600	0.532	0.527

Table 3.4 The agreement among the subjects for important sentence ranking for the evaluation set.

reference summaries were generated by ranking the sentences in the manual transcript of a spoken document by importance without assigning a score to each sentence. For the assessment of summarization performance, we adopted the widely-used ROUGE metrics [113]. It evaluates the quality of the summarization by counting the number of overlapping units, such as N -grams, longest common subsequences or skip-bigram, between the automatic summary and a set of reference summaries. Three variants of the ROUGE metrics were used to quantify the utility of the proposed methods. They are, respectively, the ROUGE-1 (unigram) metric, the ROUGE-2 (bigram) metric and the ROUGE-L (longest common subsequence) metric [113].

The summarization ratio, defined as the ratio of the number of words in the automatic (or manual) summary to that in the reference transcript of a spoken document, was set to 10%, unless otherwise stated. Since increasing the summary length tends to increase the chance of getting higher scores in the recall rate of the various ROUGE metrics and might not always select the right number of informative words in the automatic summary as compared to the reference summary, all the experimental results reported hereafter are obtained by calculating the F-scores of these ROUGE metrics. Table 3.4 shows the levels of agreement (the Kappa statistic and ROUGE metrics) between the three subjects for important sentence ranking. Each of these values was obtained by using the extractive summary created by one of the three subjects as the reference summary, in turn for each subject, while those of the other two subjects as the test summaries, and then taking their average. These observations seem to reflect the fact

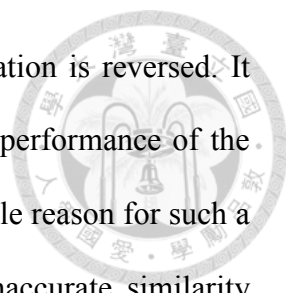
Method	Text Documents (TD)			Spoken Documents (SD)		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
ULM	0.411	0.298	0.361	0.364	0.210	0.307
VSM	0.347	0.228	0.290	0.342	0.189	0.287
LSA	0.362	0.233	0.316	0.345	0.201	0.301
MMR	0.368	0.248	0.322	0.366	0.215	0.315
MRW	0.412	0.282	0.358	0.332	0.191	0.291
LexRank	0.413	0.309	0.363	0.305	0.146	0.254
Submodularity	0.414	0.286	0.363	0.332	0.204	0.303
ILP	0.442	0.337	0.401	0.348	0.209	0.306

Table 3.5 Summarization results achieved by a few well-studied or/and state-of-the-art unsupervised methods.

that people may not always agree with each other in selecting the summary sentences for a given document.

3.2.2 Baseline Experiments

In the first place, we report on the performance level of the baseline LM-based summarization method (i.e., ULM) for extractive speech summarization by comparing it with several well-practiced or/and state-of-the-art unsupervised summarization methods, including the vector-space methods (i.e., VSM, LSA, and MMR), the graph-based methods (i.e., MRW and LexRank) and the combinational optimization methods (Submodularity and ILP). The corresponding summarization results of these unsupervised methods are illustrated in Table 3.5, where TD denotes the results obtained based on the manual transcripts of spoken documents and SD denotes the results using the speech recognition transcripts that may contain speech recognition errors. Several noteworthy observations can be drawn from Table 3.5. First, the two graph-based methods (i.e., MRW and LexRank) are quite competitive with each other and perform better than the various vector-space methods (i.e., VSM, LSA, and MMR)



for the TD case. However, for the results of the SD case, the situation is reversed. It reveals that imperfect speech recognition may adversely affect the performance of the graph-based methods as compared to vector-space methods; a possible reason for such a phenomenon is that the speech recognition errors may lead to inaccurate similarity measures between each pair of sentences. The PageRank-like procedure of the graph-based methods, in turn, will be performed based on these problematic measures, potentially leading to common results. Second, LSA representing the sentences of a spoken document to be summarized and the document itself in a low-dimensional continuous space instead of the index term (word) space, can perform slightly better than VSM in both of the TD and SD cases. Third, the Submodularity and ILP achieve the best results in the TD case, while the latter outperforms the former by a considerable margin. However, the superiority of these two methods seems to diminish for the SD case, again probably due to the effect of speech recognition errors. Fourth, the ULM method shows results that are competitive to those obtained by the other state-of-the-art unsupervised methods, which indeed justifies the viability of applying the language modeling approach for speech summarization [121][122].



Chapter 4 A Unified Framework for Pseudo-Relevance Feedback



“*Information Need*” is a key factor in information retrieval; this can be defined as “the reason for which the user turns to a search engine” [50][100]. As such, a good search engine fulfills users’ information needs. However, a critical challenge with search engines is that users typically provide scant information about their requests [18][43][173][194]. Figure 4.1 is a toy example of a user who wants to know how much the new Macbook is, and what the new specifications for the machine are.

Due to that a query usually consists of only a few words, the true query model $P(w|Q)$ might not be accurately estimated by the simple ML estimator [52]. With the alleviation of this deficiency as motivation, there are several studies devoted to estimating a more accurate query modeling, saying that it can be approached with the pseudo-relevance feedback process. Such integration seems to hold promise for query reformulation [20][54][102][190][203]. However, the success depends largely on the assumption that the set of top-ranked documents, $\mathbf{D}_{Top}=\{D_1, D_2, \dots, D_r, \dots\}$, obtained from an initial round of retrieval, are relevant and can be used to estimate a more accurate query language model.

4.1 Pseudo-Relevance Feedback

In reality, since a query often consists of only a few words, the query model that is meant to represent the user’s information need might not be appropriately estimated by the ML estimator. Furthermore, merely matching words between a query and documents

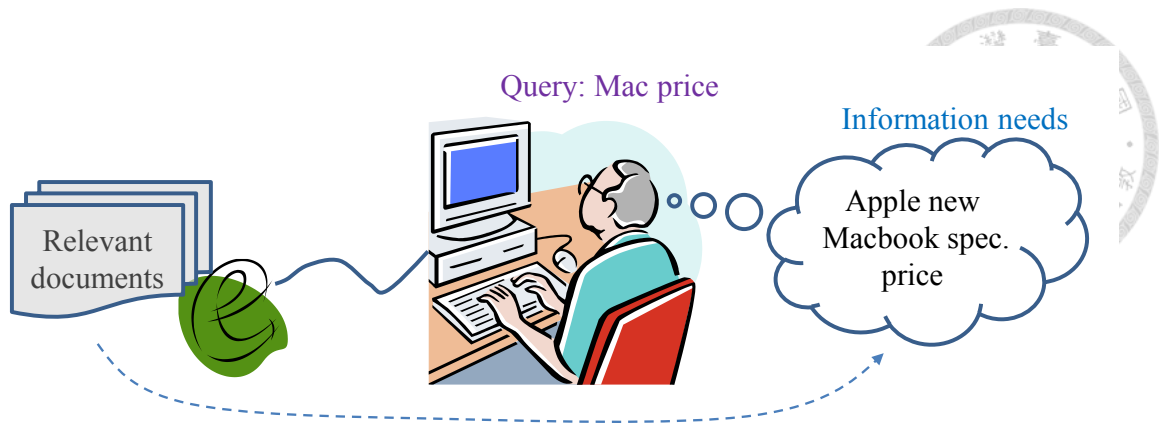


Figure 4.1 A toy example of a user goes to a search engine.

might not be an effective approach, as the word overlaps alone could not capture the semantic intent of the query. To cater for this, an LM-based SDR system with the KL-divergence measure can adopt the idea of pseudo-relevance feedback and perform two rounds of retrieval to search for more relevant documents. In the first round of retrieval, an initial query is input into the SDR system to retrieve a number of top-ranked feedback documents. Subsequently, on top of these top-ranked feedback documents, a refined query model is constructed and a second round of retrieval is conducted with this new query model and the KL-divergence measure depicted in Figure 4.2. It is usually anticipated that the SDR system can thus retrieve more documents relevant to the query.

However, an LM-based SDR system with the pseudo-relevance feedback process may confront two intrinsic challenges. One is how to purify the top-ranked feedback documents obtained from the first round of retrieval so as to remove redundant and non-relevant information. The other is how to effectively utilize the selected set of representative feedback documents for estimating a more accurate query model. For the latter, there are a number of studies proposing various query modeling techniques directly exploiting the top-ranked feedback text (or spoken) documents, such as the simple mixture model (SMM) [203], the relevance model (RM) [102] and their

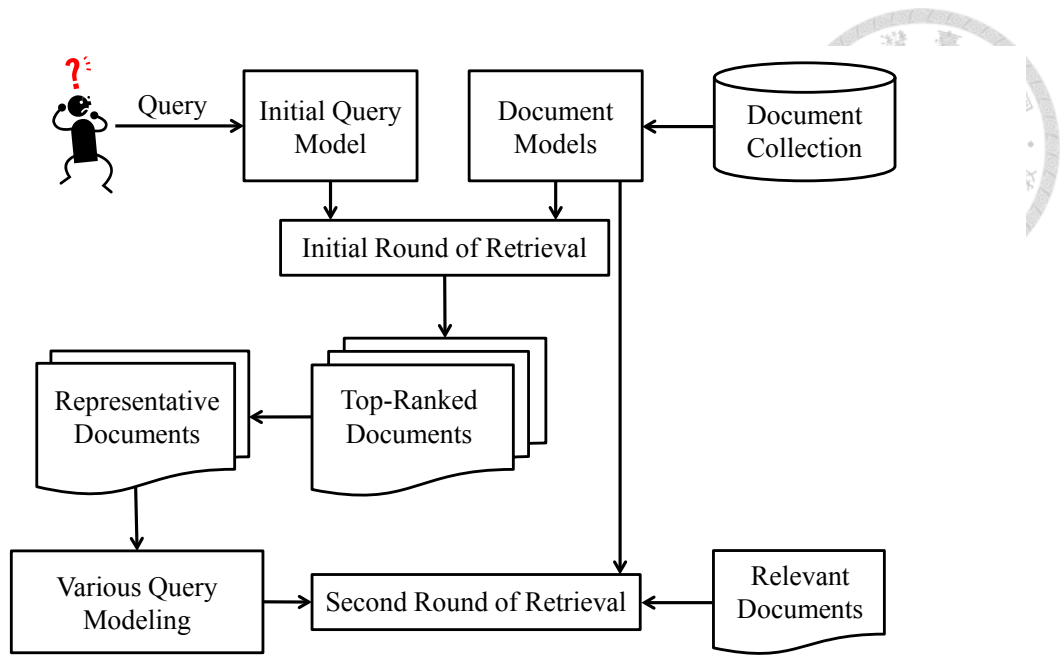
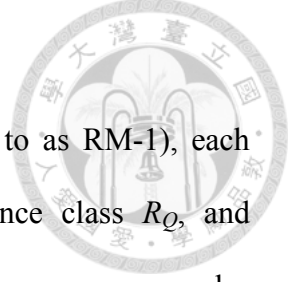


Figure 4.2 A schematic illustration of the SDR process with pseudo-relevance feedback.

extensions [43][190], among others. However, for the former, there is relatively little work done on selecting useful and representative feedback documents from the top-ranked ones for SDR, as far as we are aware. Recently, the so-called “Gapped Top K ” and “Cluster Centroid” selection methods [182] have been proposed for text information retrieval. “Gapped Top K ” selects top K documents with a ranking gap J in between any two top-ranked documents, while “Cluster Centroid” groups the top-ranked documents into K clusters and selects one representative document from each cluster to obtain diversified feedback documents. Another more attractive and sophisticated method proposed for text IR is “Active-RDD” [39][199], which takes into account the relevance, diversity and density cues of the top-ranked documents for feedback document selection.



4.1.1 Relevance Modeling (RM)

Under the notion of relevance modeling (RM, especially referred to as RM-1), each query Q is assumed to be associated with an unknown relevance class R_Q , and documents that are relevant to the semantic content expressed in query are samples drawn from the relevance class R_Q . However, in reality, since there is no prior knowledge about R_Q , we may use the top-ranked documents \mathbf{D}_{Top} to approximate the relevance class R_Q . The corresponding relevance model, on the grounds of a multinomial view of R_Q , can be estimated using the following equation [102][103]:

$$P_{RM}(w|Q) = \frac{\sum_{D_r \in \mathbf{D}_{Top}} P(D_r)P(w|D_r)\prod_{w' \in Q} P(w'|D_r)}{\sum_{D_r'' \in \mathbf{D}_{Top}} P(D_r'')\prod_{w' \in Q} P(w'|D_r'')}, \quad (4.1)$$

where the prior probability $P(D_r)$ of each document can be simply kept uniform, while the document models (such as $P(w|D_r)$) are estimated with the ML estimator on the basis of the occurrence counts of w in each document, respectively.

4.1.2 Simple Mixture Model (SMM)

Another perspective of estimating an accurate query model with the top-ranked documents is the simple mixture model (SMM), which assumes that words in \mathbf{D}_{Top} are drawn from a two-component mixture model: 1) One component is the query-specific topic model $P_{SMM}(w|Q)$, and 2) the other is a generic background model $P(w|BG)$. By doing so, the SMM model $P_{SMM}(w|Q)$ can be estimated by maximizing the likelihood over all the top-ranked documents [43][190][203]:

$$L = \prod_{D_r \in \mathbf{D}_{Top}} \prod_{w \in V} (\alpha \cdot P_{SMM}(w|Q) + (1-\alpha) \cdot P(w|BG))^{c(w, D_r)}, \quad (4.2)$$

where α is a pre-defined weighting parameter used to control the degree of reliance between $P_{SMM}(w|Q)$ and $P(w|BG)$. This estimation will enable more specific words (i.e.,

words in \mathbf{D}_{Top} that are not well-explained by the background model) to receive more probability mass, thereby leading to a more discriminative query model $P_{SMM}(w|Q)$. Simply put, the SMM model is anticipated to extract useful word usage cues from \mathbf{D}_{Top} , which are not only probably relevant to the query Q , but also external to those already captured by the generic background model.

4.1.3 Regularized Simple Mixture Model (RSMM)

Although the SMM modeling aims to extract extra word usage cues for enhanced query modeling, it may confront two intrinsic problems. One is the extraction of word usage cues from \mathbf{D}_{Top} is not guided by the original query. This would lead to a concern for SMM to be distracted from being able to appropriately model the query of interest, which is probably caused by some dominant distracting (or irrelevant) documents. The other is that the mixing coefficient α is fixed across all top-ranked documents albeit that different (either relevant or irrelevant) documents would potentially contribute different amounts of word usage cues to the enhanced query model. To mitigate these two problems, the original query model $P(w|Q)$ can be used to define a conjugate Dirichlet prior for the enhanced query model to be estimated; meanwhile, a trainable document-specific weighting coefficient α_{D_r} is introduced for each pseudo-relevant document D_r . The resulting model is referred to hereafter as the regularized simple mixture model (RSMM) and its corresponding objective likelihood function is expressed as [54][190]:

$$L = \prod_{w \in V} P_{RSMM}(w|Q)^{\mu \cdot P(w|Q)} \times \prod_{D_r \in \mathbf{D}_{Top}} \prod_{w \in V} \left(\alpha_{D_r} \cdot P_{RSMM}(w|Q) + (1 - \alpha_{D_r}) \cdot P(w|BG) \right)^{c(w, D_r)}, \quad (4.3)$$

where μ is a weighting factor indicating the confidence on the prior information (viz.

the original query model).



4.2 A Unified Framework

It is obvious that the major difference among the representative query models mentioned above is how to capitalize on the set of top-ranked documents and the original query. Taking a step forward, several subtle relationships can be deduced through the following in-depth analysis. First of all, a direct inspiration of the LM-based query reformulation framework can be drawn from the celebrated Rocchio's formulation, while the former can be viewed as a probabilistic counterpart of the latter [1][20][165][172]. The basic idea of the Rocchio's formulation is to assign higher weights to those words occurring in the top-ranked documents. Building on the same idea, the LM-based query reformulation framework has been well studied and practiced in various IR tasks and shown excellent performance. Second, after some mathematical manipulation, the formulation of the RM model (*c.f.* Eq. (4.1)) can be rewritten as:

$$P_{\text{RM}}(w|Q) = \sum_{D_r \in \mathbf{D}_{\text{Top}}} P(w|D_r) \frac{P(Q|D_r)P(D_r)}{\sum_{D_r'' \in \mathbf{D}_{\text{Top}}} P(Q|D_r'')P(D_r'')}. \quad (4.4)$$

It becomes evident that the RM model is composed by mixing a set of document models $P(w|D_r)$. The mixing coefficients are estimated by normalizing the query likelihood $P(Q|D_r)$ with respect to each pseudo-relevant document D_r , while the prior probability $P(D_r)$ of each document D_r is simply set to be uniform. As such, the RM model bears a close resemblance to the Rocchio's formulation. Furthermore, based on Eq. (4.4), we can recast the estimation of the RM model as an optimization problem, and the likelihood (or objective) function is formulated as

$$L = \prod_{w \in V} \left(\sum_{D_r \in \mathbf{D}_{Top}} P(w | D_r) P(D_r | Q) \right)^{c(w, Q)},$$

$$s.t. \quad \sum_{D_r \in \mathbf{D}_{Top}} P(D_r | Q) = 1$$



where the document models $P(w|D_r)$ are known in advance; the probability $P(D_r|Q)$ corresponding to each document D_r is unknown and leave to be estimated. Therefore, the parameters needed to be estimated are the set of mixing coefficients (i.e., $P(D_r|Q)$) and then the RM model can be formed by linearly interpolated the models of pseudo-relevant documents weighted by their corresponding coefficients. Finally, a principled framework can be obtained to unify all of these query models by using a generalized objective likelihood function:

$$L = \prod_{w \in V} \prod_{E_i \in \mathbf{E}} \left(\sum_{M_r \in \mathbf{M}} P(w | M_r) P(M_r) \right)^{c(w, E_i)},$$

$$s.t. \quad \sum_{M_r \in \mathbf{M}} P(M_r) = 1$$

where \mathbf{E} represents a set of observations which we want to maximize their likelihood, and \mathbf{M} denotes a set of mixture components.

Based on the proposed framework, here we highlight how to infer several query modeling formulations from the framework:

- **Relevance modeling:** when \mathbf{E} only consists of the user query, \mathbf{M} comprises a set of document models corresponding to the top-ranked (pseudo-relevant) documents, and we assume the document models are known, then it can be deduced to the RM model (*c.f.* Eq. (4.5)).
- **Simple mixture modeling:** if we hypothesize that \mathbf{M} consists of two components: one component is a generic background model and the other is an unknown query-specific topic model, the weight of each component is

presumably fixed in advance, and the observations are those top-ranked documents (i.e., $\mathbf{E}=\mathbf{D}_{Top}$), then we will derive the SMM model in response to the objective function (*c.f.* Eq. (4.2)).

- **Regularized simple mixture modeling:** if the weight of each component is required to be estimated as well and a Dirichlet prior is placed on the enhanced query model, the RSMM model can be obtained herewith (*c.f.* Eq. (4.3)).
- **Others:** without loss of generality, some other state-of-the-art query models also can be deduced from the proposed general objective function, such as the three-mixture model [207], the positional relevance model [128], the cluster-based methods [107][108], and among others. Furthermore, the well-practiced topic modeling [12][14][75][76] can also be deduced from the unified framework.

As a consequence, the analysis made above reveal that all of these query models bear a close resemblance to each other, and can be deduced from Eq. (4.6) with different assumptions. In the following, we will further adopt and formalize such a framework to speech recognition and summarization.

4.3 Query-specific Mixture Modeling (QMM)

The SMM model and the RSMM model are intended to extract useful word usage cues from \mathbf{D}_{Top} , which are not only relevant to the original query Q but also external to those already captured by the generic background model. However, we argue in this thesis that the “generic information” should be carefully crafted for each query due mainly to the fact that users’ information needs may be very diverse from one another. To crystallize the idea, a query-specific background model $P_Q(w|BG)$ for each query Q can

be derived from \mathbf{D}_{Top} directly. Another consideration is that since the original query model $P(w|Q)$ cannot be accurately estimated, it thus may not necessarily be the best choice for use in defining a conjugate Dirichlet prior for the enhanced query model to be estimated. As an alternative, we propose to use the RM model as a prior to guide the estimation of the enhanced query model. The enhanced query model is termed query-specific mixture model (QMM), and its corresponding training objective function can be expressed as

$$L = \prod_{w \in V} P_{QMM}(w|Q)^{\mu \cdot P_{RM}(w|Q)} \times \prod_{D_r \in \mathbf{D}_{Top}} \prod_{w \in V} (\alpha_{D_r} \cdot P_{QMM}(w|Q) + (1 - \alpha_{D_r}) \cdot P_Q(w|BG))^{c(w, D_r)}. \quad (4.7)$$

4.4 Experimental Results

Owing to each sentence S of a spoken document D to be summarized usually consists of only a few words, the corresponding sentence model $P(w|S)$ might not be appropriately estimated by the ML estimation. To alleviate the deficiency, we can leverage the merit of the above query modeling framework to estimate an accurate sentence model (or representation) for each sentence to enhance the summarization performance.

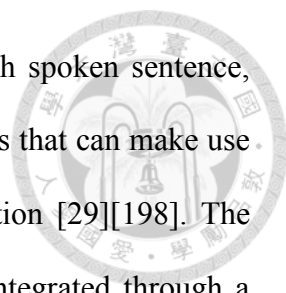
In the first part of experiments, we evaluate the utilities of the various query models as applied to the speech summarization task. At the outset, we assess the performance level of the baseline KLM method by comparison with two well-practiced unsupervised methods, viz. the vector space model (VSM) [65], and its extension, maximal marginal relevance (MMR) [19]. The corresponding results are shown in Table 4.1 and can be aligned with several related literature reviews. By looking at the results, we find that KLM outperforms VSM by a large margin, confirming the applicability of the language

	Text Documents (TD)			Spoken Documents (SD)		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
VSM	0.347	0.228	0.290	0.342	0.189	0.287
MMR	0.407	0.294	0.358	0.381	0.226	0.331
KLM	0.411	0.298	0.361	0.364	0.210	0.307
RM	0.453	0.335	0.403	0.382	0.239	0.331
SMM	0.439	0.320	0.388	0.383	0.229	0.327
RSMM	0.472	0.365	0.423	0.381	0.235	0.329
QMM	0.486	0.382	0.435	0.395	0.256	0.349
SVM	0.441	0.334	0.396	0.370	0.222	0.326
QMM +SVM	0.492	0.395	0.448	0.398	0.261	0.358

Table 4.1 The summarization results (in F-scores(%)) achieved by various language models along with text and spoken documents.

modeling framework for speech summarization. Furthermore, MMR that presents an extension of VSM performs on par with KLM for the text summarization task (TD) and exhibits superior performance over KLM for the speech summarization task (SD). We now turn to evaluate the effectiveness of the various query models (viz. RM, SMM, RSMM and QMM) in conjunction with the pseudo-relevance feedback process for enhancing the sentence model involved in the KLM method. The corresponding results are also shown in Table 4.1. Two noteworthy observations can be drawn from Table 4.1. One is that all these query models can considerably improve the summarization performance of the KLM method, which corroborates the advantage of using them for enhanced sentence representations. The other is that QMM is the best-performing one among all the formulations studied in this chapter for both the TD and SD cases.

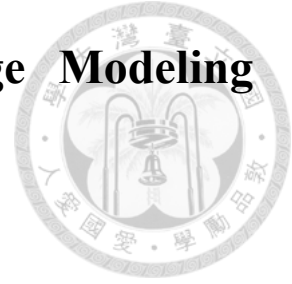
Going one step further, we explore to use extra prosodic features that are deemed complementary to the LM cue provided by QMM for speech summarization. To this end, a support vector machine (SVM) based summarization model is trained to integrate a set



of 28 commonly-used prosodic features [125] for representing each spoken sentence, since SVM is arguably one of the state-of-the-art supervised methods that can make use of a diversity of indicative features for text or speech summarization [29][198]. The sentence ranking scores derived by QMM and SVM are in turn integrated through a simple log-linear combination. The corresponding results are shown in Table 4.1, demonstrating consistent improvements with respect to all the three variants of the ROUGE metric as compared to that using either QMM or SVM in isolation. We also investigate using SVM to additionally integrate a richer set of lexical and relevance features to complement QMM and further enhance the summarization effectiveness. However, due to space limitation, we omit the details here. As a side note, there is a sizable gap between the TD and SD cases, indicating room for further improvements. We may seek remedies, such as robust indexing schemes, to compensate for imperfect speech recognition.



Chapter 5 An I-vector based Language Modeling Framework for Retrieval



Having analyzed several widely-used query models, and having proposed a principled framework to unify the relationships among them, we now propose a novel and useful language model for spoken document retrieval.

Recently, the i-vector based framework has become one of the state-of-the-art approaches for language identification (LID) [53][133][184][185] and speaker recognition (SR) [62][64][85]. One challenge of these tasks is the need to process and analyze a high-dimensional vector, which is constructed from the variable-length series of acoustic feature vectors of each input utterance based on some reference models. The i-vector framework proposed an elegant way to reduce such rough input utterance to a corresponding low-dimensional vector representation while retaining the most representative (e.g., language-specific for LID or speaker-specific for SR) information embedded in the original input utterance. Since a document is composed by a series of indexes (such as words, characters, or phonemes), our idea is to apply the i-vector framework to represent a document by a low-dimensional vector, which retains the most representative information of the document.

5.1 I-vector based Language Modeling

The i-vector framework [64][133] is a simplified variant of the joint factor analysis (JFA) approach [89][90], and both are well-known approaches for LID and SR. Their major contribution is to provide an elegant way to convert the cepstral coefficient vector

sequence of a variable-length utterance into a low-dimensional vector representation. To do so, first, a Gaussian mixture model is used to collect the Baum-Welch statistics from the utterance. Then, the first-order statistics from each mixture component are concatenated to form a high-dimensional “supervector” S , which is assumed to obey an affine linear model [89][90][185]:

$$S = \mathbf{m} + \mathbf{T} \cdot \varphi_S, \quad (5.1)$$

where \mathbf{T} is a total variability matrix, φ_S is an utterance specific latent variable, and \mathbf{m} denotes a global statistics vector. In detail, the column vectors of \mathbf{T} form a set of bases spanning a subspace covering the important variability, e.g., the language-specific evidences for LID or the speaker-specific evidences for SR, and the utterance specific variable φ_S indicates the combination of the variability of the utterance. In this way, a variable-length utterance is represented by a low-dimensional vector φ . Finally, the low-dimensional vector is applied to some well-developed post-processing techniques, such as PLDA, for LID and SR. Since the i-vector framework can be trained in an unsupervised manner while JFA must be trained along with manual annotation information, the former has become one of the state-of-the-art approaches for LID and SR recently. In this chapter, we investigate the same idea in the context of spoken document retrieval.

Specifically speaking, each document D is first represented by a high-dimensional feature vector $v_D \in \mathbb{R}^\beta$. All of the representative (e.g., lexical-, semantic-, and structure-specific) statistics are encoded in the β -dimensional vector, which obeys an affine linear model:

$$v_D = \mathbf{m} + \mathbf{T} \cdot \varphi_D, \quad (5.2)$$

where $\mathbf{T} \in \mathbb{R}^{\beta \times \gamma}$ is a total variability matrix, γ is a desired value ($\gamma \ll \beta$), and

$\mathbf{m} \in \mathbb{R}^\beta$ denotes a global statistics vector. Similarly, the column vectors of \mathbf{T} span a subspace covering the important characteristics for documents. Moreover, each document has a document specific variable $\varphi_D \in \mathbb{R}^\gamma$, which indicates the combination of the variability of the document. Based on the methodology, a disengaged version is to characterize the representative information of a document only by words. Consequently, each element of the β -dimensional vector is corresponding to a distinct word, and the probability of a word w occurring in a document D can be defined as a log-linear function:

$$P(w|D, \mathbf{T}, \mathbf{m}, \varphi_D) = \frac{\exp(\mathbf{T}_w \varphi_D + \mathbf{m}_w)}{\sum_{w' \in V} \exp(\mathbf{T}_{w'} \varphi_D + \mathbf{m}_{w'})}, \quad (5.3)$$

where \mathbf{T}_w denotes the row vector of \mathbf{T} corresponding to word w , \mathbf{m}_w denotes the statistics value of \mathbf{m} corresponding to word w , and V denotes the vocabulary inventory in the language. We name this model as the i-vector based language model (IVLM). Based on Eqs. (5.2) and (5.3), the model parameters (i.e., \mathbf{T} , φ_D and \mathbf{m}) of the proposed IVLM can be estimated by maximizing the total likelihood over all training documents:

$$L = \prod_D \prod_{w \in D} \left(\frac{\exp(\mathbf{T}_w \varphi_D + \mathbf{m}_w)}{\sum_{w' \in V} \exp(\mathbf{T}_{w'} \varphi_D + \mathbf{m}_{w'})} \right)^{c(w,D)}, \quad (5.4)$$

where $c(w,D)$ denotes the number of times the word w occurs in document D . Since estimating all the parameters jointly is intractable, we estimate them through an iterative process, i.e., we estimate \mathbf{T} and \mathbf{m} with fixed φ_D , and then estimate φ_D with fixed \mathbf{T} and \mathbf{m} :

$$\begin{aligned}
\varphi_D^{\tau+1} &= \varphi_D^\tau + \lambda \cdot \frac{\partial L}{\partial \varphi_D} \\
&= \varphi_D^\tau + \lambda \cdot \left\{ \sum_w \left[c(w, D) - |D| \cdot \frac{\exp(\mathbf{T}_w^\tau \varphi_D^\tau + \mathbf{m}_w^\tau)}{\sum_{w'} \exp(\mathbf{T}_{w'}^\tau \varphi_D^\tau + \mathbf{m}_{w'}^\tau)} \right] \mathbf{T}_w^\tau \right\}, \tag{5.5}
\end{aligned}$$

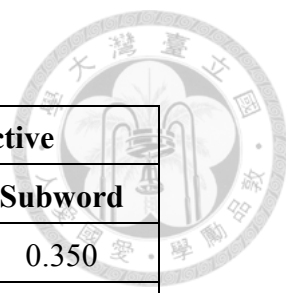
$$\begin{aligned}
\mathbf{T}_w^{\tau+1} &= \mathbf{T}_w^\tau + \lambda \cdot \frac{\partial L}{\partial \mathbf{T}_w} \\
&= \mathbf{T}_w^\tau + \lambda \cdot \left\{ \sum_D \left[c(w, D) - |D| \cdot \frac{\exp(\mathbf{T}_w^\tau \varphi_D^\tau + \mathbf{m}_w^\tau)}{\sum_{w'} \exp(\mathbf{T}_{w'}^\tau \varphi_D^\tau + \mathbf{m}_{w'}^\tau)} \right] \varphi_D^\tau \right\}, \tag{5.6}
\end{aligned}$$

$$\begin{aligned}
\mathbf{m}_w^{\tau+1} &= \mathbf{m}_w^\tau + \lambda \cdot \frac{\partial L}{\partial \mathbf{m}_w} \\
&= \mathbf{m}_w^\tau + \lambda \cdot \sum_D \left[c(w, D) - |D| \cdot \frac{\exp(\mathbf{T}_w^\tau \varphi_D^\tau + \mathbf{m}_w^\tau)}{\sum_{w'} \exp(\mathbf{T}_{w'}^\tau \varphi_D^\tau + \mathbf{m}_{w'}^\tau)} \right], \tag{5.7}
\end{aligned}$$

where λ is the step size, $|D|$ is the length of document D , and τ is the iterative index. The Frobenius norm can be used to govern \mathbf{T} and φ_D in the training process and the step size can be set empirically or by calculating the Hessian matrix [129][185].

In the retrieval phase, each document D has its own IVLM, including the document specific variable φ_D and common \mathbf{T} and \mathbf{m} . As such, the probability of word w occurring in document D computed by IVLM in Eq. (5.3) can be linearly combined with or used to replace $P(w|D)$ in the query-likelihood measure to distinguish relevant documents from irrelevant ones.

The concept of the proposed IVLM is similar to that of LSA, RLSI, and PLSA, but differences do exist among them. First, IVLM and PLSA are probabilistic models while LSA and RLSI are not. Second, IVLM not only has a different formulation to PLSA, but it does not assume that the total variability is governed by some distribution. Since the parameters of IVLM are real numbers rather than positive real numbers in PLSA, IVLM is more flexible and general than PLSA. Moreover, the parameters of IVLM can be solved in parallel while the parameters of PLSA have to be estimated in a batch mode. It




IVLM	Inductive		Transductive	
	Word	Subword	Word	Subword
short	0.336	0.360	0.382	0.350
long	0.582	0.584	0.563	0.574

Table 5.1 Retrieval results (in MAP) of IVLM with word- and subword-level index features for short and long queries using inductive and transductive learning strategies.

is worth noting that IVLM is a special (disengaged) case of the proposed i-vector based language modeling framework for SDR. We will try to discover and couple with more representative information in the future work.

5.1.1 Experimental Results

First, we will compare the use of inductive and transductive learning strategies [160] in IVLM. Inductive learning means that the models are trained from an external document collection. After training, \mathbf{T} and \mathbf{m} are used to fold-in each document d in the document collection to be retrieved to get the corresponding document specific variable φ_d . Transductive learning uses the document collection to be retrieved to train the models. After training, φ_d for each document d is used in the retrieval phase. Table 5.1 reports the retrieval results of the proposed IVLM approach for both short and long queries with respect to two learning strategies using word- or subword-level index features. We use a set of Chinese news stories from Xinhua News Agency as a contemporaneous external document set for inductive learning. It is generally believed that transductive learning should be better than inductive learning. However, as can be seen from Table 2, inductive learning achieves slightly better performance than transductive learning in

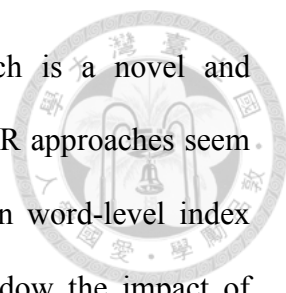


	Word		Subword	
	short	long	short	long
VSM	0.273	0.484	0.257	0.499
LSA	0.296	0.364	0.384	0.527
SCI	0.270	0.413	0.270	0.349
ULM	0.321	0.563	0.329	0.570
PLSA	0.328	0.567	0.376	0.584
LDA	0.328	0.566	0.377	0.584
IVLM	0.336	0.582	0.360	0.584

Table 5.2 Retrieval results (in MAP) of different approaches with word- and subword-level index features for short and long queries.

most cases, except when using word-level index features with short queries for SDR. Since the document collection to be retrieved (2,265 documents in total) is much smaller than the external collection (18,461 documents in total), transductive learning may suffer from the data sparseness problem while inductive learning can obtain more robust model parameters from a larger set of contemporaneous documents.

Next, the proposed IVLM approach is compared with several well-known non-probabilistic and probabilistic approaches, namely VSM, LSA, SCI, and ULM, and topic models such as PLSA and LDA. To bypass the impact of the data sparseness problem, all the approaches are trained by inductive learning. The results when using word- and subword-level index features are shown in Table 5.2. From the table, at first glance, it can be seen that the proposed IVLM framework outperforms all the non-probabilistic approaches (*c.f.* VSM, LSA, and SCI) and the probabilistic approach (*c.f.* ULM, PLSA, and LDA) in most cases. The reason why it does not perform as well with subword-level index features for short queries is not clear, and is worthy of further

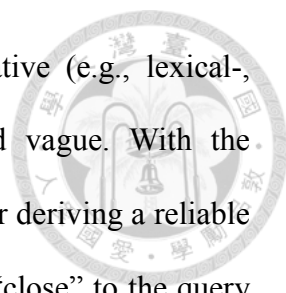


studying. The results indicate that the proposed IVLM approach is a novel and alternative way for SDR. In addition, it can also be seen that most IR approaches seem to benefit more from the use of subword-level index features than word-level index features, probably because the subword-level index units can shadow the impact of imperfect speech recognition results.

Moreover, two general observations can be made from the results. First, probabilistic approaches in general outperform non-probabilistic approaches. The results indicate that probabilistic approaches are a school of simple but powerful methods for SDR, and there are still potential research areas for non-probabilistic approaches. It should also be noticed that, the frequency count of a word is weighted by using the standard IDF method for non-probabilistic approaches while probabilistic approaches (including IVLM) only take the frequency count of a word into account. Second, a topic modeling approach outperforms its non-topic modeling counterpart (e.g., LSA vs. VSM, IVLM vs. ULM). The results indicate that the relevance between a pair of query and document should not be estimated only based on “literal term matching,” concept information is useful and should be considered in SDR.

5.2 Improved Query Representation with IVLM

An obvious deficiency inherent in the i-vector technique for both LID and SR is that, when a given speech utterance consists of only a few acoustic (cepstral coefficient) feature vectors, the low-dimensional representation learned by the i-vector technique is understandably problematic and the performance may degrade dramatically [62][71][72][85][86][91][181]. In the context of SDR, a similar deficiency occurs when we interpret a user’s information need by a low-dimensional representation, since a



query usually composes of only a few words and the representative (e.g., lexical-, semantic-, and structure-specific) statistics would be scarce and vague. With the alleviation of the scarcity problem as motivation, an intuitive idea for deriving a reliable representation for the query is to select a set of references that are “close” to the query to form a conglomerate. As such, an immediate challenge is how to determine the closeness between a candidate reference and the query. Without loss of generality, the closeness score can be one of or the combination of the degrees of acoustic, topical, semantic, syntactic, and/or literal similarities. To conjugate with the special case of the proposed IVLM model, the closeness is measured by considering only the literal similarity score (e.g., the KLM score). Similar to the scenario of applying pseudo-relevance feedback for query expansion and document re-ranking in information retrieval [28][102][190], the references are selected from the target spoken document collection. In the following, we shed light on three novel methods we propose to derive the new query representation with a set of selected references, $\mathbf{R}=\{r_1, \dots, r_{|\mathbf{R}|}\}$.

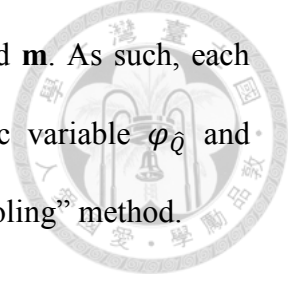
5.2.1 Sample Pooling

A straightforward way to crystallize the idea is to gather a set of selected references to form a conglomerate. Rich statistics can be mined from the conglomerate and rendered by a new β -dimensional vector $v_{\hat{Q}}$. To do so, we pool every β -dimensional vector v_{r_i} , $r_i \in \mathbf{R}$, with its closeness score to distinguish highly correlated references from less correlated references to yield a new representation, $v_{\hat{Q}}$, for a given query:

$$v_{\hat{Q}} = \alpha \cdot v_Q + (1 - \alpha) \cdot \left(\sum_{i=1}^{|\mathbf{R}|} s(Q, r_i) \cdot v_{r_i} \right), \quad (5.8)$$

where $s(Q, r_i)$ is the normalized closeness score for r_i . Finally, the query representation,

$\varphi_{\hat{Q}}$, can be derived by performing a fold-in process with $v_{\hat{Q}}$, \mathbf{T} and \mathbf{m} . As such, each query Q has its own IVLM model, including the query specific variable $\varphi_{\hat{Q}}$ and common \mathbf{T} and \mathbf{m} . We name this pooling function as the “sample pooling” method.



5.2.2 I-vector Pooling

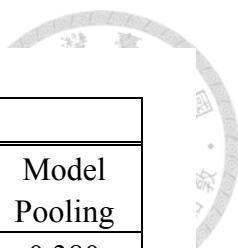
Owing to the fact that the ultimate goal of the framework is to obtain a new query representation in a low-dimensional feature vector, one reasonable type of manipulation is to craft the representation at the feature level directly. We can first interpret each reference r_i by its own representation φ_{r_i} , which is derived by performing the fold-in process with v_{r_i} , \mathbf{T} and \mathbf{m} . Then, the query representation can be obtained by pooling together all φ_{r_i} weighted by their normalized closeness scores:

$$\varphi_{\hat{Q}} = \alpha \cdot \varphi_Q + (1 - \alpha) \cdot \left(\sum_{i=1}^{|\mathbf{R}|} s(Q, r_i) \cdot \varphi_{r_i} \right). \quad (5.9)$$

We term this pooling function as the “i-vector pooling” method. Comparing the sample pooling method and the i-vector pooling method, it is evident that the former follows the original idea to enrich the statistics, based on which the new query representation is derived, while the latter composes the new query representation at the post stage directly.

5.2.3 Model Pooling

In addition to the above two pooling methods, we also propose a model-level pooling method (hereafter named “model pooling”) to derive a distributed representation for a given query:



R	Word			Subword		
	Sample Pooling	I-vector Pooling	Model Pooling	Sample Pooling	I-vector Pooling	Model Pooling
1	0.359	0.360	0.357	0.397	0.398	0.380
3	0.365	0.368	0.364	0.451	0.464	0.446
5	0.372	0.373	0.375	0.448	0.459	0.440
10	0.372	0.374	0.379	0.450	0.460	0.440
15	0.371	0.372	0.377	0.447	0.456	0.440

Table 5.3 Retrieval results (in MAP) of different pooling methods with word- and subword-level index features with respect to the number of references (**|R**).

$$P(w|\hat{Q}) = \sum_{i=1}^{|\mathbf{R}|} s(Q, r_i) \cdot P(w|r_i, \mathbf{T}, \mathbf{m}, \varphi_{r_i}), \quad (5.10)$$

where $P(w|r_i, \mathbf{T}, \mathbf{m}, \varphi_{r_i})$ designates the corresponding IVLM model of reference r_i .

5.2.4 Experimental Results

In the set of experiments, we evaluate the capability of IVLM to enhance query representation in SDR. The results when using different pooling methods (i.e., the sample pooling, i-vector pooling, and model pooling) and different levels of index units, as well as different numbers of references, are shown in Table 5.3. It is worth noting that, KLM is equivalent to QLM when the query model is simply estimated by an empirical ML estimator. Thus, the baseline performance here is equivalent to that of QLM shown in Table 5.2. Several observations can be drawn from Table 5.3. First, it is clear that the proposed framework outperforms the baseline KLM model (*c.f.* QLM in Table 5.2) in all cases. This indicates that IVLM is able to improve the estimation of the query model for better document ranking in SDR. Second, all the proposed pooling methods have comparable performance, and they outperform all of the retrieval models compared in

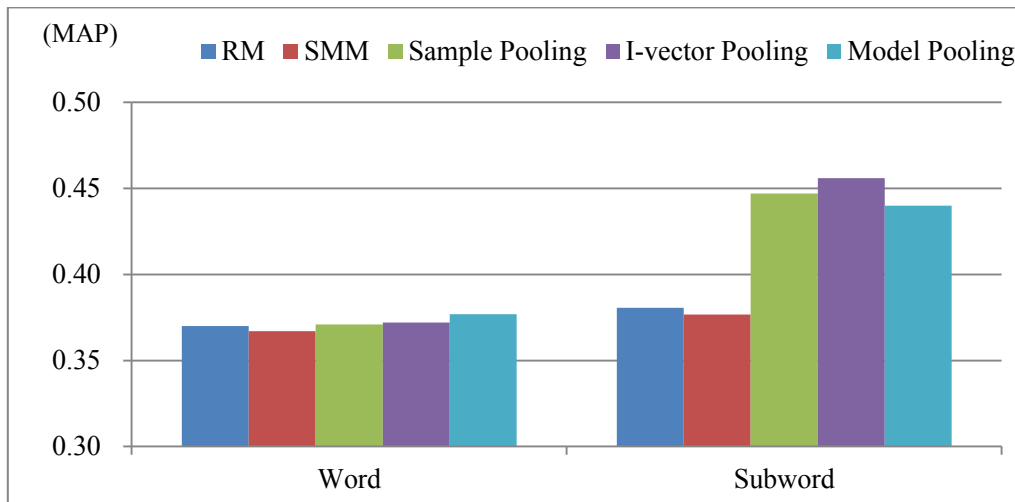


Figure 5.1 Retrieval results (in MAP) of i-vector based query representation techniques, relevance model (RM), and simple mixture model (SMM) with word- and subword-level index features.

Table 5.2. Third, the experimental results indicate that the best setting of the number of references is around 5~10 for the word-level index features and 3 for the subword-level index features. Comparing the results in Table 3 with that of IVLM in Table 5.2, it can be seen that accurate query modeling seems to be more crucial to the retrieval performance than enhanced document modeling. A reason might be that a document is usually long enough for building a reliable representation while an accurate query representation is usually much harder to be inferred from a short query. Moreover, it can also be seen that most retrieval models seem to benefit from the use of subword-level index features, probably because the subword-level index units can shadow the impact of imperfect speech recognition results to some extent.

Next, we further compare the proposed framework with two representative LM-based methods for query reformulation [35], namely the relevance model (RM) and the simple mixture model (SMM), which have been well-practiced and proved their capability in various text IR tasks. The number of the pseudo-relevant documents for RM and SMM

(and the references respectively for the proposed three IVLM-based methods) is set to 15. The corresponding retrieval results with different levels of index units are depicted in Figure 5.1. The results indicate that all of these models deliver comparable performance when using word-level index features, while the proposed three IVLM-based query models outperform the two representative query models by a big margin when using subword-level index units. The reason might be that the model parameters are more accurately estimated, since the observations will increase when fine-grained index units are used to index queries and documents. In sum, the marked results have confirmed that IVLM indeed is efficient and effective for representing queries and documents in SDR.



Chapter 6 A RNNLM-based Framework for Summarization



While the bag-of-words assumption makes ULM a clean and efficient method for sentence ranking, it is an oversimplification of the problem of extractive speech summarization. Intuitively, long-span context dependence (or word proximity) cues might provide an additional indication of the semantic-relatedness of a given sentence with regard to the document to be summarized. Although a number of studies had been done on extending ULM to further capture local context dependence simply based on n -grams of various orders (e.g., bigrams or trigram), most of them resulted in leading to mild gains or mixed results [38]. This is due in large part to the fact that a sentence usually consists of only a few words and the complexity of the n -gram model increases exponentially with the order n , making it difficult to obtain reliable probability estimates with the ML criterion. In view of such phenomena, we explore a novel recurrent neural network language modeling (RNNLM) framework for the formulation of the sentence models involved in the LM-based summarization approach.

6.1 Recurrent Neural Network Language Modeling for Speech Summarization

RNNLM has recently emerged as a promising modeling framework that can effectively and efficiently render the long-span context relationships among words (or more precisely, the dependence between an upcoming word and its whole history) for use in speech recognition [138][139][140]. The fundamental network of RMMLM is

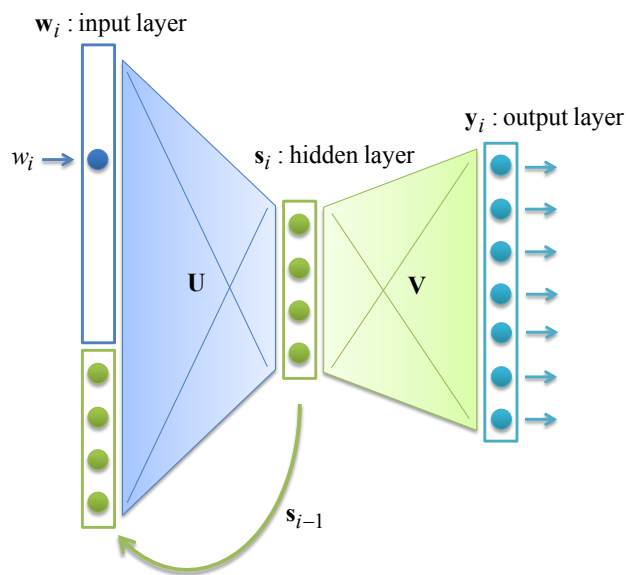


Figure 6.1 A schematic depiction of the fundamental network of RNNLM.

schematically depicted in Figure 6.1, which consists of three main ingredients: the input layer, the hidden layer and the output layer. For each time index i , the input vector \mathbf{w}_i is in one-of- V encoding, indicating the currently encountered word w_i , where the vector size V is set equal to the number of distinct vocabulary words; the hidden vector \mathbf{s}_i represents the statistical cues encapsulated thus far in the network for the history (i.e., all preceding words) of w_i ; and the output layer vector \mathbf{y}_i stores the predicted likelihood values for each possible succeeding word (or word class) of w_i . An attractive aspect of RNNLM is that the statistical cues of previously encountered word retained in the hidden layer, i.e., \mathbf{s}_{i-1} , can be fed back to the input layer and work in combination with the currently encountered word w_i as an “augmented” input vector for predicting an arbitrary succeeding word w_{i+1} . By doing so, RNNLM can naturally take into account not only word usage cues but also long-span structural information of word co-occurrence relationships for language modeling. A bit of terminology: the augmented input vector \mathbf{x}_i , the hidden vector \mathbf{s}_i and the output vector \mathbf{y}_i are, respectively,

represented or computed as follows [138][139][140]:

$$\mathbf{x}_i = [(\mathbf{w}_i)^T, (\mathbf{s}_{i-1})^T]^T, \quad (6.1)$$

$$\mathbf{s}_i = f(\mathbf{U}\mathbf{x}_i), \quad (6.2)$$

$$\mathbf{y}_i = g(\mathbf{V}\mathbf{s}_i), \quad (6.3)$$

where $f(\cdot)$ and $g(\cdot)$ are pre-defined sigmoid activation functions and softmax functions, respectively. Finally, the model parameters (i.e., \mathbf{U} and \mathbf{V}) of RNNLM can be derived by maximizing the likelihood of the training corpus using the back-propagation through time (BPTT) algorithm [16][80][162] that virtually unfolds the feedback loop of RNNLM making its model structure bear a close resemblance to the family of so-called deep neural networks [112] and thereby learn to remember word usage information for several time steps encapsulated in the hidden layer of RNNLM [6][138].

As the notion of RNNLM is adopted and formalized for sentence modeling in extractive speech summarization, we devise a hierarchical training strategy to obtain the corresponding RNNLM model for each sentence of a spoken document to be summarized, which proceeds in three stages:

1. First of all, a document-level RNNLM model is trained for each document to be summarized by using the document itself as the training data. The resulting RNNLM model will memorize not only word usage but also long-span word dependence cues inherent in the document.
2. After that, for each sentence of the spoken document to be summarized, the corresponding sentence-specific RNNLM model is trained, starting from the document-level RNNLM model obtained in Stage 1 and using the sentence itself as the adaptation data for model training. That is, the parameters of RNNLM are optimized by maximize the likelihood of the sentence.



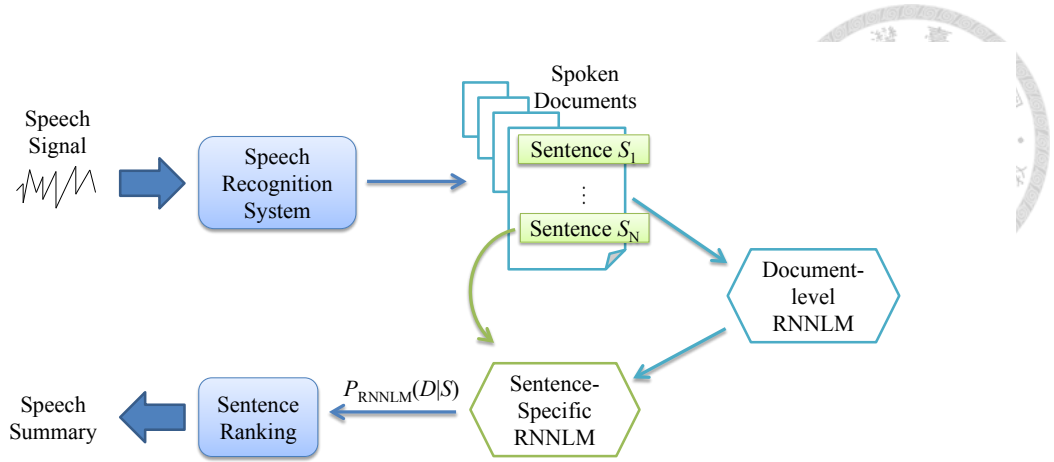
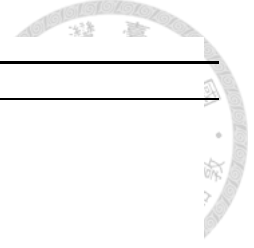


Figure 6.2 A sketch of the proposed RNNLM summarization framework.

3. Consequently, the resulting sentence-specific RNNLM model can be used in place of, or to complement, the original sentence model (i.e., ULM). The RNNLM-based sentence generative probability for use in sentence ranking can be computed by

$$P_{\text{RNNLM}}(D | S) = \prod_{i=1}^L P_{\text{RNNLM}}(w_i | w_1, \dots, w_{i-1}, S). \quad (6.4)$$

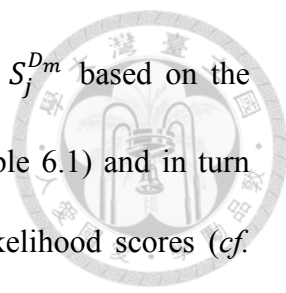
A schematic illustration of the proposed RNNLM-based summarization framework is depicted in Figure 6.2, while a highlight of the corresponding model training and important sentence ranking procedures is given in Table 6.1. In the following, we elaborate on some important steps involved in Table 6.1. 1) In the initial phase, a desired number of the hidden layer neurons H of each RNNLM and a set of documents \mathbf{D} to be summarized, where each document D_m in \mathbf{D} contains $|D_m|$ sentences (each of which is represented by $S_j^{D_m}$), are given. 2) Then, in the training phase, since the architecture of the prototype RNNLM model is a three-layer neural network, there are two sets of parameters (i.e., \mathbf{U}_m and \mathbf{V}_m) for each document D_m to be summarized, which are estimated using the back-propagation through time (BPTT) algorithm (*cf.* Line 3 in Table 6.1). Following that, the model parameters of the sentence-level RNNLM model



Input:	
H : Number of Hidden Layer Neurons	
$\mathbf{D} = \{D_1, \dots, D_m, \dots, D_M\}$ $D_m = \{S_1^{D_m}, \dots, S_j^{D_m}, \dots, S_{ D_m }^{D_m}\}$	
Model Training & Important Sentence Ranking:	
1:	for D_1 to D_M do
2:	document-level RNNLM model training
3:	$\mathcal{L}(\mathbf{U}_m, \mathbf{V}_m) = \sum_{i=1}^{ D_m } \log(y_i)$
4:	for $S_1^{D_m}$ to $S_{ D_m }^{D_m}$ do
5:	sentence-level RNNLM model training
6:	$\mathcal{L}(\mathbf{U}_{S_j^{D_m}}, \mathbf{V}_{S_j^{D_m}} \mathbf{U}_m, \mathbf{V}_m) = \sum_{i=1}^{ S_j^{D_m} } \log(y_i)$
7:	end for
8:	for $S_1^{D_m}$ to $S_{ D_m }^{D_m}$ do
9:	calculate document likelihood
10:	$P(D_m S_j^{D_m}) = \prod_{i=1}^{ S_j^{D_m} } P(w_i w_1, \dots, w_{i-1}, S_j^{D_m})$
11:	$= \prod_{i=1}^{ S_j^{D_m} } P(w_i \mathbf{U}_{S_j^{D_m}}, \mathbf{V}_{S_j^{D_m}}, S_j^{D_m})$
12:	end for
13:	Sentence selection according to $P(D_m S_j^{D_m})$
14:	end for

Table 6.1 Training of RNNLM-based sentence models and the application of them for important sentence ranking.

(i.e., $\mathbf{U}_{S_j^{D_m}}$ and $\mathbf{V}_{S_j^{D_m}}$) for each sentence $S_j^{D_m}$ in D_m is estimated starting from the document-level model parameters (i.e., \mathbf{U}_m and \mathbf{V}_m) of D_m obtained from previous step (cf. Line 6 in Table 6.1). 3) Finally, in the important sentence ranking phase, we can

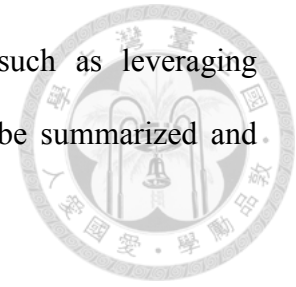


calculate the document likelihood score offered by each sentence $S_j^{D_m}$ based on the corresponding RNNLM model of $S_j^{D_m}$ (cf. Lines 10 and 11 in Table 6.1) and in turn select important sentences of D_m according their the document likelihood scores (cf. Line 13 in Table 6.1). Interested readers may also refer to [93][105][123] for more in-depth discussions on a number of efficient training algorithms developed for RNNLM.

It should be noticed that the training strategy described above can also be viewed as an instantiation of curriculum learning [9][57], which seeks to apply a specific and well-planned ordering of the training data for estimating machine-learning models (such as neural networks) to be better suited for a target application. However, as far as we are aware, there is still not much research on leveraging RNNLM along with the aforementioned curriculum-learning strategy for extractive speech summarization. In this thesis, we also make a step further by analyzing and comparing the effectiveness of the RNNLM-based summarization methods with other well-practiced state-of-the-art methods thoroughly.

Also worth mentioning is that there has been an alternative realization of the LM approach to extractive summarization that exploits the KL-divergence to measure, for example, the discrepancy of the word (unigram) distribution in a candidate sentence and that in the original document for important (summary) sentence ranking [69][116]. With some algebraic manipulations, it is easy to show that the effect of the KL-divergence for important sentence ranking is negatively equivalent to the document likelihood (document unigram probability) generated by the sentence $P(D|S)$ (i.e., the ULM method), once the document model is estimated merely on the basis of the empirical frequency of words in the document. However, it seems to be more straightforward to

extend ULM with higher-order language modeling strategies, such as leveraging RNNLM to measuring the relatedness between the document to be summarized and each of its sentences.



6.2 Experimental Results

6.2.1 Experiments on Higher-order N -gram and Topic Language Modeling

In the second set of experiments, we first investigate a simple extension of the ULM method by using a bigram language model smoothed with a unigram language model to represent each sentence involved in a document to be summarized (denoted by BLM hereafter). As elaborated before, the weakness of the ULM method lies in that it follows the strict bag-of-words assumption (an oversimplification) without considering the word regularity or proximity information within spoken documents. The corresponding summarization results achieved by the BLM method are depicted in Table 6.2. To our surprise, the integration of bigram and unigram cues together (i.e., BLM) for sentence modeling only arrives at almost the same performance level as that using the unigram information alone (i.e., ULM) in the SD case, but performs even worse than the latter in the TD case. A reasonable explanation is that the estimation of the bigram language model for each sentence inevitably suffers from a more serious data sparseness problem than the unigram model, since its number of model parameters would be at most the square of that of the latter. As a side note, we have also experimented on using a trigram language model, smoothed with both unigram and bigram language models, to represent each spoken sentence; however, it delivered almost negligible improvements over the ULM and BLM methods.

Method	Text Documents (TD)				Spoken Documents (SD)			
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4
ULM	0.411	0.298	0.362	0.300	0.361	0.215	0.311	0.214
BLM	0.411	0.298	0.362	0.300	0.361	0.215	0.311	0.214
PLSA	0.382	0.260	0.350	0.266	0.327	0.188	0.284	0.189
PLSA+ULM	0.433	0.317	0.379	0.320	0.378	0.234	0.332	0.226
RNNLM	0.433	0.319	0.390	0.319	0.330	0.184	0.294	0.180
RNNLM+ULM	0.533	0.439	0.483	0.430	0.439	0.304	0.393	0.289

Table 6.2 Summarization results achieved by various LM-based methods, including ULM, BLM, PLSA, PLSA+ULM, RNNLM and RNNLM+ULM.

Instead of constructing the sentence models based on literal term information (such as the statistics of word unigrams or bigrams), we also exploit probabilistic topic models to represent sentences through a latent topic space. For example, each sentence of a spoken document to be summarized is interpreted as a probabilistic latent semantic analysis (PLSA) model [75][204] consisting of a set of K shared latent topics $\{T_1, \dots, T_k, \dots, T_K\}$ with sentence-specific topic weights $P(T_k|S)$, while each topic offers a unigram (multinomial) distribution $P(w_i|T_k)$ for observing an arbitrary word w_i of the vocabulary:

$$P_{\text{PLSA}}(D|S) = \prod_{i=1}^L [\sum_{k=1}^K P(w_i|T_k)P(T_k|S)], \quad (6.5)$$

where the probability $P(w_i|T_k)$ can be estimated beforehand based on a large set of text or speech documents, while the probability $P(T_k|S)$ of each sentence can be estimated on-the-fly during the summarization process using the expectation-maximization (EM) algorithm [204]. The resulting sentence-specific PLSA model can be used in isolation (denoted by PLSA), or in linear combination with the unigram language model (denoted by PLSA+ULM), to compute the sentence generative probability for important sentence selection. As indicated in Table 6.2, PLSA alone cannot match the performance of ULM,

Method	Ratio	Text Documents (TD)			Spoken Documents (SD)		
		ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
ULM	10%	0.411	0.298	0.361	0.364	0.210	0.307
	20%	0.483	0.368	0.420	0.428	0.255	0.355
	30%	0.551	0.432	0.481	0.471	0.304	0.399
RNNLM +ULM	10%	0.533	0.439	0.483	0.439	0.304	0.393
	20%	0.580	0.478	0.522	0.491	0.341	0.428
	30%	0.639	0.540	0.574	0.514	0.354	0.445

Table 6.3 Summarization results respectively achieved by ULM and RNNLM+ULM with respect to different summarization ratios.

largely because PLSA only offers coarse-grained concept clues about the sentences at the expense of losing discriminative power among concept-related words in finer granularity. On the other hand, the combination of PLSA with ULM (PLSA+ULM) can lead to noticeable improvements as compared to that using either PLSA or ULM alone.

6.2.2 Experiments on the Proposed RNNLM Summarizer

In the third set of our experiments, we evaluate the effectiveness of the proposed RNNLM method for extractive speech summarization. The deduced sentence-specific RNNLM model can be used in isolation (denoted by RNNLM), or linearly combined with the unigram language model (denoted by RNNLM+ULM), to compute the sentence generative probability; the corresponding results are shown in Table 6.2 as well. In order to verify the utility of RNNLM and RNNLM+ULM in capturing long-distance word co-occurrence relationships (especially when compared to the other LM-based methods), we additionally include the summarization results evaluated with the ROUGE-SU4 (skip-bigram with maximum gap length of 4) metric in Table 6.2 [113]. ROUGE-SU4 is a frequently-used metric for summarization performance evaluation, which quantifies the degree of overlap between the reference and automatically

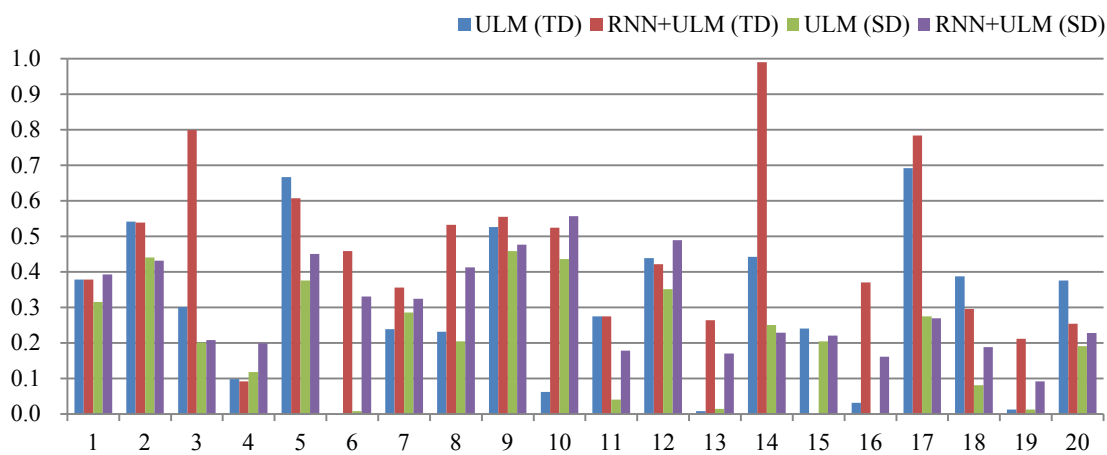
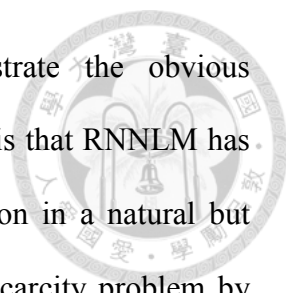


Figure 6.3 Summarization results (in ROUGE-2) for each individual document (represented with either manual or speech transcript) in the test set, respectively, achieved by ULM and RNNLM+ULM.

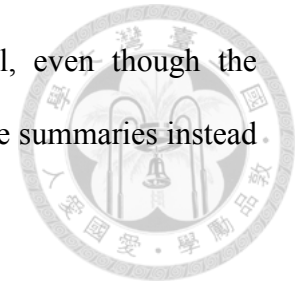
generated summaries in terms of not only unigrams but also distant skip-bigrams.

Comparing to the existing LM-based methods (i.e., ULM BLM, PLSA and PLSA+ULM) or the subcategories of unsupervised methods (*c.f.* Table 3.5), we can find that RNNLM+ULM consistently and significantly surpasses all the other models in both the TD and SD cases; however, using RNNLM in isolation only leads to improved results in the TD case. Furthermore, two more particularities can be made when we look into the results of Table 6.2. On one hand, because RNNLM+ULM manages to encapsulate not only word usage cues but also long-distance word co-occurrence relationships for sentence modeling, it seems to perform particularly well when the evaluation metrics are based on counting the number of matched high-order word co-occurrence counts between the reference and automatically generated summaries, such as the ROUGE-2, ROUGE-L and ROUGE-SU4 metrics. On the other hand, RNNLM and ULM seem to be complementary of each other and indeed can conspire to obtain better sentence modeling. Furthermore, when we compare RNNLM (or



RNNLM+ULM) with BLM, the experimental results demonstrate the obvious superiority of RNNLM that might be attributed to two causes. One is that RNNLM has the inherent advantage for capturing long-span structure information in a natural but systematic way. The other is that RNNLM can mitigate the data scarcity problem by implicitly performing clustering of words aside their histories (or preceding words) into a lower-dimensional continuous space, which makes the language model prediction (or probability calculation) based on such compact representations of words aside their histories become more robust [139][141]. One thing to note is that we have also tried to combine ULM, PLSA and RNNLM together for achieving better summarization accuracy; however, such an attempt only leads to roughly comparable performance as RNNLM+ULM. It is thus believed that the way to systemically combine these models is still a challenging issue and needs further in-depth investigation and proper experimentation. Figure 6.3 depicts the summarization results (in ROUGE-2) for each individual document (represented with either manual or speech transcript) in the test set, achieved by ULM and RNNLM+ULM. A closer look at these results also reveals that RNNLM+ULM can indeed boost the performance of ULM significantly for most of the test documents that are more difficult to be summarized (for example, Documents 6, 13, 16 and 19 in the test set). In order to further assess the quality of the automatically generated summaries of our RNNLM-based methods and the other state-of-the-art methods compared in this thesis, we also take an additional set of abstractive summaries written by the same three human subjects as the ground truth for performance evaluation. For this purpose, the human subjects were instructed to do human summarization, respectively, by writing an abstract for each document with a length (in words) being roughly 25% of the original broadcast news story. The corresponding results are shown in Table 6.4, which indicate that RNNLM+ULM can provide

consistent and significant gains over the other methods as well, even though the reference summaries being used are the human-generated abstractive summaries instead of the human-generated extractive summaries.



6.2.3 More Empirical Analysis of the RNNLM Summarizer

To gain more insights into the merit of the RNNLM-based summarization framework, we additionally conduct empirical performance analysis on the RNNLM summarizer from three different aspects. First, we assess the statistical significance of the improvements that are delivered by RNNLM+ULM over ULM with the Student's paired t -test, which confirms that RNNLM+ULM indeed significantly outperforms ULM (with the p -values smaller than 0.005 for both the TD and SD cases). Second, to further confirm such superiority of RNNLM+ULM over ULM, we also conduct speech summarization with different summarization ratios (i.e., 20% or 30%), in addition the default setting of 10%; the corresponding results are shown in Table 6.3. It is evident that RNNLM+ULM consistently leads to marked improvements over ULM for summarization ratios of 20% and 30%, in terms of all the three ROUGE metrics; significance tests, again, indicate the statistical significance of such improvements. Third, we turn to investigate the impact of the model complexity of RNNLM (more specifically, the number of hidden neurons being used) on the ultimate summarization performance. As revealed by results shown in Table 6.5, using a small number of hidden neurons (i.e., 25 or 50) seems to be adequate for the speech summarization task studied here. This can be attributed to the fact that since each sentence of a spoken document to be summarized usually consists of only a few words, the RNNLM model of each sentence, which has smaller complexity, tends to have more reliable estimation of its model parameters. Nevertheless, the way to systemically determine the optimal number

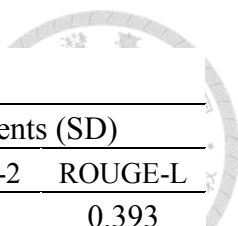
Method	Text Documents (TD)			Spoken Documents (SD)		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
ULM	0.375	0.231	0.314	0.348	0.178	0.286
VSM	0.325	0.175	0.262	0.325	0.161	0.264
LSA	0.315	0.152	0.254	0.303	0.139	0.243
MMR	0.344	0.193	0.289	0.348	0.182	0.285
MRW	0.381	0.226	0.316	0.342	0.183	0.283
LexRank	0.312	0.173	0.262	0.281	0.120	0.227
Submodularity	0.394	0.235	0.334	0.336	0.188	0.295
ILP	0.368	0.234	0.317	0.313	0.158	0.268
PLSA+ULM	0.389	0.245	0.327	0.359	0.193	0.299
RNNLM	0.337	0.218	0.297	0.337	0.218	0.297
RNNLM+ULM	0.423	0.281	0.362	0.369	0.218	0.316

Table 6.4 Summarization results achieved by the proposed framework and a few well-studied or/and state-of-the-art unsupervised methods, which were measured by using the abstractive summaries written by the human subjects as the ground truth.

of hidden-layer neurons of RNNLM for each spoken document to be summarized remains an open issue and needs further investigation. On the other hand, we have also experimented on deepening the architecture of our RNNLM model to be a four-layer network [93], which was in turn used to couple with our proposed training strategy for the modeling of each spoken sentence. Unfortunately, such a deeper RNNLM architecture only yielded mixed summarization results as compared to the three-layer RNNLM architecture we adopted in this chapter.

6.2.4 Further Extensions on RNNLM Summarizer

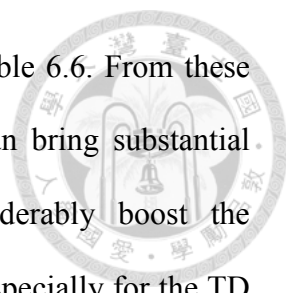
A potential downside of our proposed RNNLM-based summarization framework is that the resulting summarizer performs important sentence ranking and selects the top-ranked sentences to form a summary simply based on (in decreasing order of) the



Number of Neurons	Text Documents (TD)			Spoken Documents (SD)		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
25	0.526	0.436	0.474	0.439	0.304	0.393
50	0.533	0.439	0.483	0.432	0.296	0.385
100	0.465	0.359	0.474	0.426	0.289	0.373
150	0.492	0.386	0.439	0.407	0.263	0.358
200	0.428	0.310	0.376	0.425	0.281	0.374

Table 6.5 Summarization results achieved by RNNLM+ULM with respect to different numbers of hidden-layer neurons being used.

relevance measure between a spoken document to be summarized and each sentence in the document (namely, the likelihood that the RNNLM+ULM (or RNNLM alone) model of each sentence generates the document), without taking into account the relationships among sentences. However, it is generally expected that a desirable summary should not only include highly topic-relevant sentences as many as possible, but at the same time try to reduce the redundancy among these selected sentences as much as possible. To remedy this situation, we further explore to integrate the relevance measure provided by RNNLM+ULM into other state-of-the-art unsupervised summarizers that simultaneously consider the issues of topic coverage and redundancy removal during the summarization process. Here we take MMR [19] and ILP [135] as two examples for the purpose of exploration. For MMR, we use the RNNLM+ULM based measure to replace the original cosine similarity measure involved in the iterative selection process of MMR (denoted by RNNLM+ULM+MMR). On the other hand, for ILP, the RNNLM+ULM based measure is employed not only to compute the importance (relevance) weights between any pair of the document to be summarized and one of its sentences, but also to estimate the redundancy degree involved in the constrained combinational optimization process of ILP (denoted by



RNNLM+ULM+ILP). Their corresponding results are shown in Table 6.6. From these results, it is obvious that these two simple integrated methods can bring substantial gains to MMR and ILP, respectively, while they also considerably boost the summarization performance of using RNNLM+ULM in isolation, especially for the TD case. These results again corroborate the intuition that a good extractive summary should contain relevant and diverse sentences that cover the main topics or aspects of an original spoken document.

6.2.5 RNNLM with Syllable-level Index Units

In an attempt to mitigate the summarization performance degradation caused by imperfect speech recognition, we explore to make possible use of subword-level index units for the proposed RNNLM-based methods. To do this, syllable pairs are taken as the basic units for indexing instead of words. The recognition transcript of each spoken document, in form of a word stream, was automatically converted into a stream of overlapping syllable pairs. Then, all the distinct syllable pairs occurring in the spoken document collection were then identified to form a vocabulary of syllable pairs for indexing. We can thus use the syllable pairs (as a surrogate of words) to represent the spoken documents and sentences, and subsequently construct the associated summarization models of disparate methods based on such representations. The corresponding results for both the TD and SD cases, achieved by ULM, RNNLM and RNNLM+ULM in conjunction with syllable-level index units, are shown in Table 6.7. We may draw attention to two observations here. First, the results, in general, have consistent trends with the previous sets of experiments where the documents are indexed with words (*c.f.* Table 6.2). Second, the subword-level (syllable-level) index units seem to show comparable or even better performance than the word-level index

Method	Text Documents (TD)			Spoken Documents (SD)		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
RNNLM+ULM	0.533	0.439	0.483	0.439	0.304	0.393
MMR	0.368	0.248	0.322	0.366	0.215	0.315
ILP	0.442	0.337	0.401	0.348	0.209	0.306
RNNLM+ULM+MMR	0.538	0.450	0.489	0.445	0.312	0.395
RNNLM+ULM+ILP	0.554	0.465	0.505	0.444	0.312	0.399

Table 6.6 Summarization results achieved by RNNLM+ULM, MMR, ILP and their combinations.

Method	Text Documents (TD)			Spoken Documents (SD)		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
ULM	0.533	0.439	0.483	0.439	0.304	0.393
RNNLM	0.368	0.248	0.322	0.366	0.215	0.315
RNNLM+ULM	0.558	0.337	0.401	0.348	0.209	0.306

Table 6.7 Summarization results achieved by ULM, RNNLM and RNNLM+ULM in conjunction with syllable-level index features.

units (*c.f.* Table 6.1) when being used with the RNNLM-based methods for performing summarization with imperfect speech recognition transcripts (i.e., for the SD case). We conjecture this is because subword-level index units work more robustly against speech recognition errors and the out-of-vocabulary problem, thus likely leading to better summarization performance.

6.2.6 Coupling RNNLM with Extra Acoustic Features

In the final set of experiments, we explore the potential of extracting extra acoustic features inherent in spoken sentences for use in summarization. To this end, we use a set of sixteen indicative features crafted based on four commonly-used types of acoustic values, as outlined in Table 6.8, to characterize a spoken sentence. In implementation, the acoustic features were extracted from the spoken sentences using the Praat toolkit

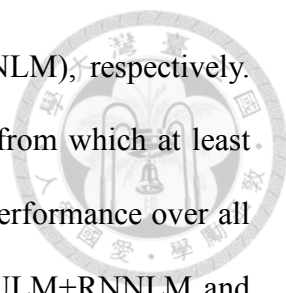
1. Pitch Value (min, max, diff, avg.)
2. Peak Normalized Cross-correlation of Pitch Value (min, max, diff, avg.)
3. Energy Value (min, max, diff, avg.)
4. Duration Value (min, max, diff, avg.)

Table 6.8 Four types of acoustic features used to represent each spoken sentence.

Method	Spoken Documents (SD)			
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-SU4
SVM(AC)	0.373	0.235	0.332	0.220
SVM(AC+ULM)	0.378	0.236	0.335	0.224
SVM(AC+RNNLM)	0.387	0.250	0.344	0.239
SVM(AC+ULM+RNNLM)	0.407	0.268	0.363	0.255

Table 6.9 Summarization results achieved by using acoustic features in isolation and its combination with ULM, RNNLM and ULM+RNNLM based sentence ranking scores, respectively.

[15]. Interested readers may refer to [117] for detailed accounts on the characteristics of these features and comparisons among them. Here SVM is chosen as the exemplar summarizer to integrate these derived acoustic features (i.e., taking them as the input that represents each sentence) for important spoken sentence ranking. The corresponding model was trained beforehand with the development set in a supervised manner, and the resulting SVM summarizer is denoted by SVM(AC) hereafter. Furthermore, we also study to take the ranking score of ULM, RNNLM and ULM+RNNLM implemented with syllable-level index units, respectively, as an additional indicative feature fed into SVM to represent each sentence (note that the score corresponds to the normalized document likelihood in the logarithmic domain, predicted by the respective sentence generative model), leading to an augmented set of seventeen features in total. The resulting SVM summarizers are denoted by



SVM(AC+ULM), SVM(AC+RNNLM), and SVM(AC+ULM+RNNLM), respectively. Table 6.9 shows the results of these summarizers for the SD case, from which at least two observations can be drawn. First, SVM(AC) exhibits superior performance over all the unsupervised summarizers compared in this thesis, except for ULM+RNNLM and its variants (*cf.* Tables IV, V and IX). Unlike the unsupervised summarizers, SVM(AC), however, requires human annotation in the training phase. Second, SVM(AC+ULM), SVM(AC+RNNLM), and SVM(AC+ULM+RNNLM) all yield better performance than SVM(AC). Although SVM(AC+ULM+RNNLM) stands out in performance among these SVM-based summarizers, to our surprise, it does not in general operate as effectively as ULM+RNNLM and its variants (implemented with either word- or syllable-level index units). This means that the way to systemically combine the acoustic features with other indicative features (especially those seemingly superior-performing ones) for important spoken sentence selection remains a challenging issue and needs further in-depth investigation and proper experimentation.



Chapter 7 A Word Embedding Framework for Summarization



In language model research, recurrent neural networks represent a breakthrough in building language models; recently, research trends have moved from *modeling* to *vectorization*. Several representation learning approaches have been proposed and applied to various NLP-related tasks.

7.1 Classic Word Embedding Methods

Perhaps one of the most-known seminal studies on developing word embedding methods was presented in [8]. It estimated a statistical (N -gram) language model, formalized as a feed-forward neural network, for predicting future words in context while inducing word embeddings (or representations) as the by-product. Such an attempt has already motivated many follow-up extensions to develop similar methods for probing latent semantic and syntactic regularities in various NLP applications. Representative methods may include, but are not limited to, the continuous bag-of-words (CBOW) model [141], the skip-gram (SG) model [141][144] and the global vector (GloVe) model [164]. However, as far as we are aware, there is little work done to contextualize these methods for use in speech summarization.

7.1.1 Continuous Bag-of-Words (CBOW) Model

Rather than seeking to learn a statistical language model, the continuous bag-of-words model manages to obtain a dense vector representation (embedding) of each word

directly [141]. The structure of CBOW is similar to a feed-forward neural network, with the exception that the non-linear hidden layer in the former is removed. By doing so, the model can still retain good performance and be trained on much more data efficiently while getting around the heavy computational burden incurred by the non-linear hidden layer. Formally, given a sequence of words, w^1, w^2, \dots, w^T , the objective function of CBOW is to maximize the log-probability expressed as follows:

$$\sum_{t=1}^T \log P(w^t | w^{t-c}, \dots, w^{t-1}, w^{t+1}, \dots, w^{t+c}), \quad (7.1)$$

where c is the window size of the training context for the central word w^t ; T denotes the length of the training corpus. The conditional probability in Eq. (7.1) is defined by

$$P(w^t | w^{t-c}, \dots, w^{t-1}, w^{t+1}, \dots, w^{t+c}) = \frac{\exp(\mathbf{v}_{\bar{w}^t} \cdot \mathbf{v}_{w^t})}{\sum_{i=1}^V \exp(\mathbf{v}_{\bar{w}^t} \cdot \mathbf{v}_{w_i})}, \quad (7.2)$$

where \mathbf{v}_{w^t} denotes the vector representation of the word w at position t ; V indicates the size of the vocabulary; and $\mathbf{v}_{\bar{w}^t}$ denotes the (weighted) average of the vector representations of the context words of w^t [141][166], which can be further expressed in the form

$$\mathbf{v}_{\bar{w}^t} = \sum_{j=-c, j \neq 0}^c \alpha_j \mathbf{v}_{w^{t+j}}, \quad (7.3)$$

where α_j is a weighting factor associated with the distance between the central word w^t and the context word w^{t+j} . The concept of CBOW is motivated by the distributional hypothesis [144], which states that words with similar meanings often occur in similar contexts and thus suggests to look for word representations that capture their context distributions.

7.1.2 Skip-Gram (SG) Model

In contrast to the CBOW model, the SG model employs an inverse training objective for

learning word representations with a simplified feed-forward neural network [141][142]. Formally, given a sequence of words, w^1, w^2, \dots, w^T , the objective function of SG is to maximize the following log-probability:

$$\sum_{t=1}^T \sum_{j=-c, j \neq 0}^c \log P(w^{t+j} | w^t), \quad (7.4)$$

where c is the window size of the training context for the central word w^t ; and the conditional probability can be calculated by

$$P(w^{t+j} | w^t) = \frac{\exp(\mathbf{v}_{w^{t+j}} \cdot \mathbf{v}_{w^t})}{\sum_{i=1}^V \exp(\mathbf{v}_{w_i} \cdot \mathbf{v}_{w^t})}, \quad (7.5)$$

where $\mathbf{v}_{w^{t+j}}$ and \mathbf{v}_{w^t} denote the word representations of words at position $t+j$ and t , respectively. In implementation of CBOW and SG, the hierarchical soft-max algorithm (HSM) [142][149] and the negative sampling algorithm (NS) [142][147] have been introduced to make the training of model parameters more efficient and effective.

7.1.3 Global Vector (GloVe) Model

The GloVe model suggests that an appropriate starting point for word representation learning should be associated with the ratios of co-occurrence probabilities rather than the predict probabilities [164]. More precisely, GloVe makes use of a weighted least squares regression, which aims at learning word representations by preserving the co-occurrence frequencies between each pair of words:

$$\sum_{i=1}^V \sum_{j=1}^V f(X_{w_i w_j}) (\mathbf{v}_{w_i} \cdot \mathbf{v}_{w_j} + b_{w_i} + b_{w_j} - \log X_{w_i w_j})^2, \quad (7.6)$$

where $X_{w_i w_j}$ denotes the number of times word w_i and w_j co-occurs in a pre-defined sliding context window; $f(\cdot)$ is a monotonic smoothing function used to modulate the impact of each pair of words involved in the model training; and \mathbf{v}_w and b_w denote the word representation and the bias term of word w , respectively.



7.1.4 Analytic Comparisons

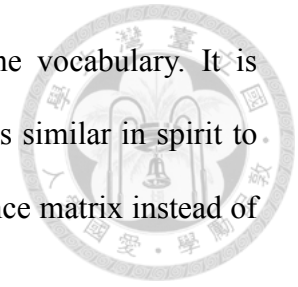
There are several analytic comparisons can be made among the above three word embedding methods. First, they have different model structures and learning strategies. CBOW and SG adopt an on-line learning strategy, i.e., the parameters (word representations) are trained sequentially. Therefore, the order that the training samples are used may change the resulting models dramatically. In contrast, GloVe uses a batch learning strategy, i.e., it accumulates the statistics over the entire training corpus and updates the model parameters at once. Second, it is worthy to note that SG (trained with the negative sampling algorithm) and GloVe have an implicit/explicit relation with the classic weighted matrix factorization approach, while the major difference is that SG and GloVe concentrate on rendering the word-by-word co-occurrence matrix but weighted matrix factorization is usually concerned with decomposing the word-by-document matrix [65][111][32].

The observations made above on the relation between word embedding methods and matrix factorization bring us to the notion of leveraging the singular value decomposition (SVD) method as an alternative mechanism to derive the word embeddings in this chapter. Given a training text corpus, we have a word-by-word co-occurrence matrix \mathbf{A} . Each element \mathbf{A}_{ij} of \mathbf{A} is the log-frequency of times words w_i and w_j co-occur in a pre-defined sliding context window in the corpus. Subsequently, SVD decomposes \mathbf{A} into three sub-matrices:

$$\mathbf{A} \approx \mathbf{U} \Sigma \mathbf{V}^T = \tilde{\mathbf{A}}, \quad (7.7)$$

where \mathbf{U} and \mathbf{V} are orthogonal matrices, and Σ is a diagonal matrix. Finally, each row vector of matrix \mathbf{U} (or the column vector of matrix \mathbf{V}^T , $\mathbf{U}=\mathbf{V}$ since \mathbf{A} is a symmetric

matrix) designates the word embedding of a specific word in the vocabulary. It is worthy to note that using SVD to derive the word representations is similar in spirit to latent semantic analysis (LSA) but using the word-word co-occurrence matrix instead of the word-by-document co-occurrence matrix [1].



7.2 Leveraging Word Embeddings for Summarization

The original goal of word embedding techniques is to represent each word by a continuous distributed (or distributional) vector. However, in the context of extractive speech summarization, the similarity between each pair of document and one of its sentences should be determined, and an extractive summary can thus be generated on top of the similarity measure being adopted.

7.2.1 Cosine Similarity Measure

The vector space model (VSM) [205] has long been the basis for many NLP-related tasks, including text or speech summarization. The major advantage of VSM is that it is simple and intuitive, while being efficient and effective. In VSM, each document (and sentence) is represented by a high-dimensional vector, where each dimension specifies the occurrence statistics associated with an index term (e.g., word, subword, or their n -grams) in the document (and sentence). The relevance degree between a pair of sentence and document is estimated by the cosine measure of their vector representations.

Motivated by the idea of VSM, a straightforward way to leverage the word embedding methods for speech summarization is to represent a candidate summary sentence (and a document to be summarized) by averaging all the representations of the

words occurring in the sentence (or the document) [84]; here we take CBOW and the sentence representation as an example:

$$\mathbf{v}_S = \sum_{w \in S} \frac{n(w, S)}{|S|} \mathbf{v}_w. \quad (7.8)$$

By doing so, the document D to be summarized and each sentence S of the document will have a fixed-length dense vector representation; their relevance degree can be computed by the cosine similarity measure:

$$SIM(S, D) = \frac{\mathbf{v}_S \cdot \mathbf{v}_D}{\|\mathbf{v}_S\| \cdot \|\mathbf{v}_D\|}. \quad (7.9)$$

Therefore, important sentences can be ranked in decreasing order of this measure and in turn be selected to form a summary to represent the original spoken document. The notion of pairing word embedding methods with the cosine similarity measure has recently attracted much attention and been applied with success to many NLP-based applications. However, as far as we are aware, this notion has never been extensively explored in extractive speech summarization.

7.2.2 The Triplet Learning Model

Inspired by the vector space model (VSM), a straightforward way to leverage the word embedding methods for extractive SDS is to represent a sentence S_i (and a document D to be summarized) by averaging the vector representations of words occurring in the sentence S_i (and the document D) [84][189]:

$$\mathbf{v}_{S_i} = \sum_{w \in S_i} \frac{n(w, S_i)}{|S_i|} \mathbf{v}_w. \quad (7.10)$$

By doing so, the document D and each sentence S_i of D will have a respective fixed-length dense vector representation, and their relevance degree can be evaluated by

the cosine similarity measure.

However, such an approach ignores the inter-dimensional correlation between two vector representations. To mitigate the deficiency of the cosine similarity measure, we employ a triplet learning model to enhance the estimation of the similarity degree between a pair of representations [10][21][49][152]. Without loss of generality, our goal is to learn a similarity function, $R(\cdot, \cdot)$, which assigns higher similarity scores to summary sentences than to non-summary sentences, i.e.,

$$R(\mathbf{v}_D, \mathbf{v}_{S_i}^+) > R(\mathbf{v}_D, \mathbf{v}_{S_j}^-) \quad (7.11)$$

where $\mathbf{v}_{S_i}^+$ denotes the sentence representation (in the form of a column vector) for a summary sentence S_i , while $\mathbf{v}_{S_j}^-$ is the representation for a non-summary sentence S_j .

The parametric ranking function has a bi-linear form as follows:

$$R(\mathbf{v}_D, \mathbf{v}_S) \equiv \mathbf{v}_D^T \mathbf{W} \mathbf{v}_S, \quad (7.12)$$

where $\mathbf{W} \in \mathbb{R}^{d \times d}$, and d is the dimension of the vector representation. By applying the passive-aggressive learning algorithm presented in [34], we can derive the similarity function R such that all triplets obey

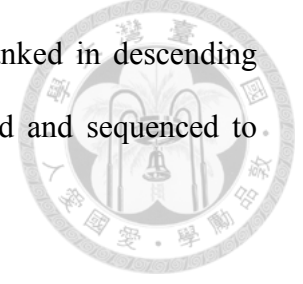
$$R(\mathbf{v}_D, \mathbf{v}_{S_i}^+) > R(\mathbf{v}_D, \mathbf{v}_{S_j}^-) + \delta. \quad (7.13)$$

That is, not only the similarity function will distinguish summary and non-summary sentences, but also there is a safety margin of δ between them. With δ , a hinge loss function can be defined as

$$loss(\mathbf{v}_D, \mathbf{v}_{S_i}^+, \mathbf{v}_{S_j}^-) = \max\{0, \delta - R(\mathbf{v}_D, \mathbf{v}_{S_i}^+) + R(\mathbf{v}_D, \mathbf{v}_{S_j}^-)\}. \quad (7.14)$$

Then, \mathbf{W} can be obtained by applying an efficient sequential learning algorithm

iteratively over the triplets [21][49]. With \mathbf{W} , sentences can be ranked in descending order of similarity measure, and the top sentences will be selected and sequenced to form a summary according to a target summarization ratio.



7.2.3 Document Likelihood Measure

When applying the LM-based approach to extractive speech summarization, a principal realization is to use a probabilistic generative paradigm for ranking each sentence S of a spoken document D to be summarized, which can be expressed by $P(S|D)$. Stimulated by the document likelihood measure adopted by the ULM method, for the various word representation methods studied here, we investigate to first construct a word-based language model for predicting the occurrence probability of other word w_j . Taking CBOW as an example, the probability of a word w_j given another word w_i can be calculated by:

$$P(w_j | w_i) = \frac{\exp(\mathbf{v}_{w_j} \cdot \mathbf{v}_{w_i})}{\sum_{w_l \in V} \exp(\mathbf{v}_{w_l} \cdot \mathbf{v}_{w_i})}. \quad (7.15)$$

During the summarization process, we can linearly combine the associated word-based language models of the words occurring in a sentence S to form a composite sentence-specific language model for S . Consequently, the document likelihood measure can be computed by:

$$P(D | S) = \prod_{w_j \in D} \left[\sum_{w_i \in S} \alpha_{w_i} \cdot P(w_j | w_i) \right]^{n(w_j, D)}, \quad (7.16)$$

where the weighting coefficient α_{w_i} is set to be in proportion to the frequency of w_i occurring in sentence S and is summed to 1 (i.e., $\sum_{w_i \in S} \alpha_{w_i} = 1$). The sentences offering the highest document likelihoods can be selected and sequenced to form the

final summary according to different summarization ratios.



7.2.4 Experimental Results

We now turn to investigate the utilities of three state-of-the-art word embedding methods (i.e., CBOW, SG, and GloVe) and the proposed SVD method (*c.f.* Section 7.1.4), working in conjunction with the cosine similarity measure for speech summarization. The results are shown in Table 7.1. From the results, several observations can be made. First, the three state-of-the-art word embedding methods (i.e., CBOW, SG, and GloVe), though with disparate model structures and learning strategies, achieve comparable results to each other in both the TD and SD cases. Although these methods outperform the conventional VSM model, they only achieve almost the same level of performance as LSA and MMR, two improved versions of VSM, and perform worse than MRW, LexRank, SM, and ILP in the TD case. To our surprise, the proposed SVD method outperforms other word embedding methods by a substantial margin in the TD case and slightly in the SD case. It should be noted that the SVD method outperforms not only CBOW, SG, and GloVe, but also LSA and MMR. It even outperforms all the methods compared in Table 3.5 in the SD case.

In the next set of experiments, we evaluate the capability of the triplet learning model for improving the measurement of similarity when applying word embedding methods in speech summarization. The results are shown in Table 7.2. From the table, two observations can be drawn. First, it is clear that the triplet learning model outperforms the baseline cosine similarity measure (*c.f.* Table 7.1) in all cases. This indicates that triplet learning is able to improve the measurement of the similarity degree for sentence ranking and considering the inter-dimensional correlation in the similarity measure

Method	Text Documents (TD)			Spoken Documents (SD)		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
CBOW	0.369	0.224	0.308	0.365	0.206	0.313
SG	0.367	0.230	0.306	0.358	0.205	0.303
GloVe	0.367	0.231	0.308	0.364	0.214	0.312
SVD	0.409	0.265	0.342	0.374	0.215	0.319

Table 7.1 Summarization results achieved by various word-embedding methods in conjunction with the cosine similarity measure.

Method	Text Documents (TD)			Spoken Documents (SD)		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
CBOW	0.472	0.367	0.432	0.396	0.258	0.347
SG	0.404	0.284	0.348	0.374	0.223	0.321
GloVe	0.372	0.248	0.315	0.375	0.225	0.319
SVD	0.422	0.303	0.364	0.376	0.223	0.323

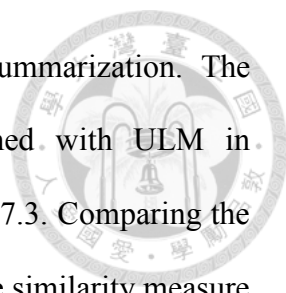
Table 7.2 Summarization results achieved by various word-embedding methods in conjunction with the triplet learning model.

Method	Text Documents (TD)			Spoken Documents (SD)		
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L
CBOW	0.456	0.342	0.398	0.385	0.237	0.333
SG	0.436	0.320	0.385	0.371	0.225	0.322
GloVe	0.422	0.309	0.372	0.380	0.239	0.332
SVD	0.411	0.298	0.361	0.364	0.222	0.313

Table 7.3 Summarization results achieved by various word-embedding methods in conjunction with the document likelihood measure.

between two vector representations is indeed beneficial. Second, “CBOW with triplet learning” outperforms all the methods compared in Table 1 in both the TD and SD cases. However, we have to note that learning \mathbf{W} has to resort to a set of documents with reference summaries; thus the comparison is unfair since all the methods in Table 3.5 are unsupervised ones.

In the last set of experiments, we pair the word embedding methods with the

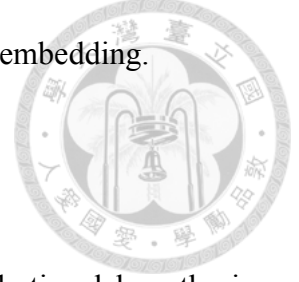


document likelihood measure for extractive spoken document summarization. The deduced sentence-based language models were linearly combined with ULM in computing the document likelihood. The results are shown in Table 7.3. Comparing the results to that of the word embedding methods paired with the cosine similarity measure (*c.f.* Table 7.1), it is evident that the document likelihood measure works pretty well as a vehicle to leverage word embedding methods for speech summarization. We also notice that CBOW outperforms the other three word embedding methods in both the TD and SD cases, just as it had done previously in Table 7.2 when combined with triplet learning, whereas “SVD with document likelihood measure” does not preserve the superiority as “SVD with triplet learning” (*c.f.* Table 7.2). Moreover, comparing the results with that of various state-of-the-art methods (*c.f.* Table 3.5), the word embedding methods with the document likelihood measure are quite competitive in most cases.

7.3 Probabilistic Word Embeddings

Although the aforementioned methods for learning representations of words have enjoyed success in capturing latent semantic and syntactic regularities (relationships) among words, the interpretation about the learned word representations, however, remains somewhat opaque. In view of this, we propose a novel learning framework instantiated with various modeling structures to infer appropriate representations of words for speech summarization, which has a clear and rigorous probabilistic interpretation. We will begin with some terminology about the framework. Let \mathbf{M} denotes a $D \times V$ matrix, where the i -th column of the matrix \mathbf{M} corresponds to the word representation for the i -th word w_i in the vocabulary. Let \mathbf{W} denotes a $D \times V$ matrix, where the j -th column of the matrix \mathbf{W} is the word representation for the target

word w_j in the vocabulary. D is a pre-defined dimension of the word embedding.



7.3.1 Probabilistic Bag-of-Words (PBOW) Model

The first word embedding method is also stimulated by the distributional hypothesis [144], as stated previously in Section 7.1.1. The fundamental notion of this model is to learn appropriate representations of words that can facilitate better prediction of an arbitrary target word given that some of its surrounding context words are observed. To crystallize such an idea, the objective function is defined to maximize the total likelihood over all vocabulary words occurring in the training corpus:

$$L = \prod_{i=1}^V P(w_i | C_i)^{n(w_i)} = \prod_{i=1}^V \left(\frac{\mathbf{W}_{w_i}^T \cdot \mathbf{H}_{w_i}}{\sum_{j=1}^V \mathbf{W}_{w_j}^T \cdot \mathbf{H}_{w_i}} \right)^{n(w_i)}, \quad (7.17)$$

where $n(w_i)$ is the frequency count of word w_i occurring in the training corpus, C_i denotes the context information collected from the entire corpus for word w_i , and \mathbf{H}_{w_i} is the corresponding vector representation for C_i :

$$\mathbf{H}_{w_i} = \sum_{t=1}^T \sum_{k=-c, k \neq 0}^c I[w^t = w_i] \alpha_{t,k} \mathbf{M}_{w^{t+k}}, \quad (7.18)$$

where $I[x]$ is designated as an indicator function whose output value is 1 if the statement x is true and 0 otherwise; $\alpha_{t,k}$ is a weighting factor associated with the distance between the central word w^t and its surrounding context word w^{t+k} , and is summed to 1 (i.e., $\sum_{t=1}^T \sum_{k=-c, k \neq 0}^c I[w^t = w_i] \alpha_{t,k} = 1$). To make the computation more efficient, we assume each word w corresponds to a multinomial (distributional) representation (i.e., each column vector of the matrix, denoted by \mathbf{M}_w). In essence, we can think of such a distributional representation as a special case of the conventional distributed representation. Moreover, we further assume that each “row” vector of matrix \mathbf{W} is a multinomial distribution as well. Consequently, the objective function of the model can

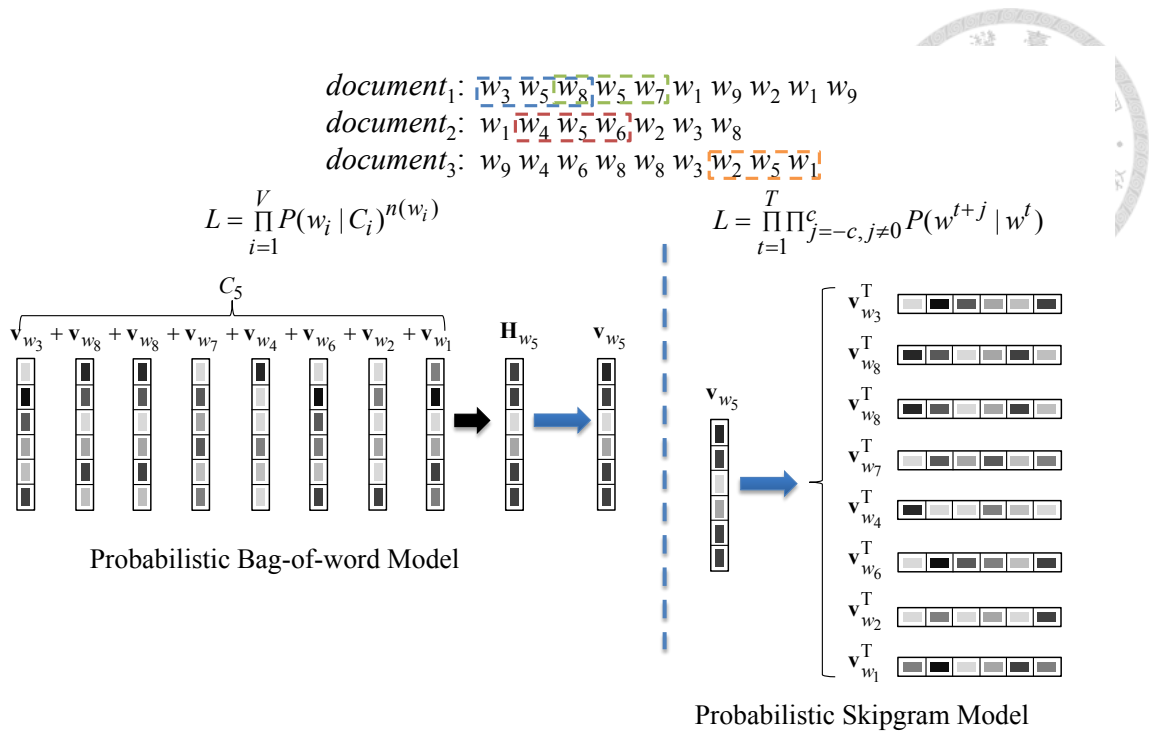


Figure 7.1 A running toy example for learning disparate distributional representations of a specific word w_5 , where the training corpus contains three documents, the vocabulary size is 9 (i.e., having words w_1, \dots, w_9) and the context window size is 1 (i.e., $c=1$).

be simplified as follows:

$$L = \prod_{i=1}^V P(w_i | C_i)^{n(w_i)} = \prod_{i=1}^V (\mathbf{W}_{w_i}^T \cdot \mathbf{H}_{w_i})^{n(w_i)}. \quad (7.19)$$

Obviously, such a model bears close resemblance to CBOW and can be viewed as a probabilistic counterpart of CBOW. Below, we refer to it as the probabilistic bag-of-words (PBOW) model, while its component distributions (i.e., \mathbf{W} and \mathbf{M}) can be estimated using the expectation-maximization (EM) algorithm [52].

7.3.2 Probabilistic Skip-gram (PSG) Model

In contrast to the PBOW model, which learns the representation of each word w_i through estimating the probability distributions of its context words that collectively

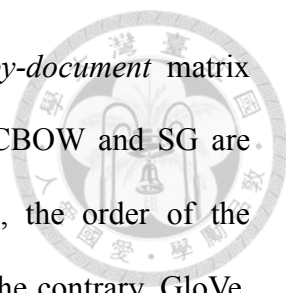
generate w_i , an alternative approach is to obtain an appropriate word representation by considering the predictive ability of a given word occurring at an arbitrary position of the training corpus (denoted by w^t) to predict its surrounding context words. For the idea to go, we define the objective function of such a model as

$$\begin{aligned}
 L &= \prod_{t=1}^T \prod_{j=-c, j \neq 0}^c P(w^{t+j} | w^t) \\
 &= \prod_{t=1}^T \prod_{j=-c, j \neq 0}^c \frac{\mathbf{W}_{w^{t+j}}^T \cdot \mathbf{M}_{w^t}}{\sum_{k=1}^V \mathbf{W}_{w_k}^T \cdot \mathbf{M}_{w^t}}.
 \end{aligned} \tag{7.20}$$

Again, since we assume that each column of \mathbf{W} is a multinomial distribution, the terms in the denominator will be summed to one and thus we can omit the denominator here. This model is similar in spirit to SG and can be regarded as a probabilistic counterpart of SG. As such, we will term the resulting model as the probabilistic skip-gram model (PSG) hereafter. Following a similar vein to PBOW, the component distributions of PSG can also be estimated with the EM algorithm. A running example for the proposed two models is schematically depicted in Figure 7.1.

7.3.3 Analytic Comparisons

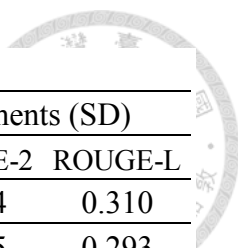
CBOW, SG, GloVe, PBOW, and PSG can be analyzed from several critical perspectives. First, the training objectives for all of these models aim at maximizing the collection likelihood, but their respective update formulations are different. The model parameters of CBOW, SG and GloVe are updated by variants of the stochastic gradient descent-based (SGD) algorithm [55][142], while PBOW and PSG are estimated by the expectation-maximization (EM) algorithm. It is worthy to note that GloVe has a close relation with the classic weighted matrix factorization approach, while the major difference is that the former concentrates on rendering the *word-by-word* co-occurrence



matrix and the latter is concerned with decomposing the *word-by-document* matrix [32][65]. Second, since the parameters (word representations) of CBOW and SG are trained sequentially (i.e., the so-called *on-line* learning strategy), the order of the training corpus may affect their resulting models dramatically. On the contrary, GloVe, PBOW and PSG accumulate the statistics over the entire training corpus in the first place; the corresponding model parameters of these models are then updated based on such censuses at once (i.e., the so-called *batch-mode* learning strategy). Finally, due to that in our models (PBOW and PSG) we assume each row of the matrix \mathbf{W} is designated as a multinomial distribution, the by-product is that the columns of \mathbf{W} collectively can be thought of as forming a latent semantic space whose meaning can be explained by the component multinomial distributions. Therefore, word representations learned by our proposed models (i.e., \mathbf{M}) can readily be realized and interpreted by referring to the matrix \mathbf{W} . More formally, the word vectors learned by PBOW and PSG are distributional representations, while CBOW, SG and GloVe present each word by a distributed representation. To the best of our knowledge, this is the first study of such an interpretation when learning word representations.

7.3.4 Experimental Results

We now turn to investigate the utilities of three state-of-the-art word embedding methods (i.e., CBOW, SG and GloVe) and our proposed methods (i.e., PBOW and PSG), respectively working in conjunction with the cosine similarity measure for speech summarization. The corresponding results are shown in Table 7.4, where HSM denotes the condition when the model parameters were obtained based on the hierarchal soft-max algorithm, while NS denotes learning with the negative sampling algorithm. Several observations can be made from the experimental results. First, all the three



Method	Text Documents (TD)			Spoken Documents (SD)			
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L	
GloVe	0.366	0.244	0.310	0.363	0.214	0.310	
CBOW	HSM	0.360	0.199	0.294	0.357	0.185	0.293
	NS	0.359	0.200	0.293	0.363	0.193	0.300
SG	HSM	0.370	0.209	0.305	0.346	0.180	0.283
	NS	0.366	0.211	0.306	0.345	0.179	0.282
PBOW	0.397	0.283	0.346	0.376	0.233	0.326	
PSG	0.403	0.281	0.351	0.380	0.234	0.330	

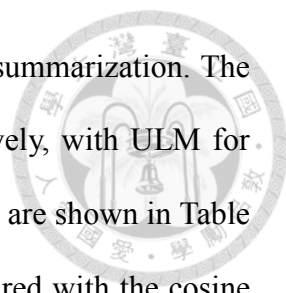
Table 7.4 Summarization results achieved by various word-embedding methods in conjunction with the cosine similarity measure.

Method	Text Documents (TD)			Spoken Documents (SD)			
	ROUGE-1	ROUGE-2	ROUGE-L	ROUGE-1	ROUGE-2	ROUGE-L	
GloVe	0.422	0.309	0.372	0.380	0.239	0.332	
CBOW	HSM	0.472	0.364	0.417	0.372	0.226	0.316
	NS	0.456	0.342	0.398	0.385	0.237	0.333
SG	HSM	0.436	0.323	0.385	0.372	0.223	0.323
	NS	0.436	0.320	0.385	0.371	0.225	0.322
PBOW	0.437	0.331	0.387	0.386	0.241	0.332	
PSG	0.434	0.333	0.389	0.375	0.244	0.331	

Table 7.5 Summarization results achieved by various word-embedding methods in conjunction with the document likelihood measure.

state-of-the-art word embedding methods, though based on disparate model structures and learning strategies, achieve results competitive to each other for both TD and SD cases. Albeit that these methods outperform the conventional VSM model, they achieve almost the same level of performance as LSA and MMR, which are considered to be two enhanced versions of VSM (*c.f.* Table 3.5). It should be noted that the proposed methods not only outperform than CBOW, SG and GloVe, but also are better than LSA and MMR for most of the TD and SD cases (*c.f.* Table 3.5).

In the next set of experiments, we evaluate the various word embedding methods



paired with the document likelihood measure for extractive speech summarization. The deduced sentence-based language models were combined, respectively, with ULM for computing document likelihoods [202] and the corresponding results are shown in Table 7.5. Comparing to the results of these word embedding methods paired with the cosine similarity measure (*c.f.* Table 7.4), it is evident that the document likelihood measure seems be a preferable vehicle to leverage word-embedding methods for speech summarization. As we look into the detailed results of Table 7.5, we notice two particularities. On one hand, CBOW seems to perform better than others in the TD case, whereas the superiority does not seem to preserve in the SD case. On the other hand, if we compare the results with that of the other state-of-the-art summarization methods (*c.f.* Table 3.5), the word embedding methods with the document likelihood measure still outperform them by a margin for most of the TD and SD cases.



Chapter 8 Conclusion and Outlook



Language model research can be dated back to the n -gram language model which is blocked by the frequency counts of words and the multinomial distributions. The original goal of the n -gram language model was to determine the probability of a given word sequence. Following this model, several researchers then proposed a variety of architectures for language models that capturing fine- or coarse-grained semantic and syntactic regularities. The wide array of language models that have been developed so far fall roughly into four main categories: 1) word-regularity models, 2) topic models, 3) continuous language models, and 4) neural network-based language models (*c.f.* Chapter 2.1). Founded on a variety of pioneering research, this thesis has proposed several novel extensions, described new developments, and shared interesting findings. Figure 8.1 summarizes some important language models year by year, and also summarizes the contributions of this thesis.

■ The Unified Framework for Pseudo-Relevance Feedback

Language models have been widely used for information retrieval. However, this approach has two major challenges: 1) a query is often a vague expression of the underlying information need, and 2) there can be word usage mismatch between a query and a document even if they are topically related to each other. To mitigate these problems, in Chapter 4, we reformulated the original queries using relevance-based language models using different objective functions, and then proposed a principled framework to unify the relationships among most of the widely-used query modeling formulations. The school of research has also

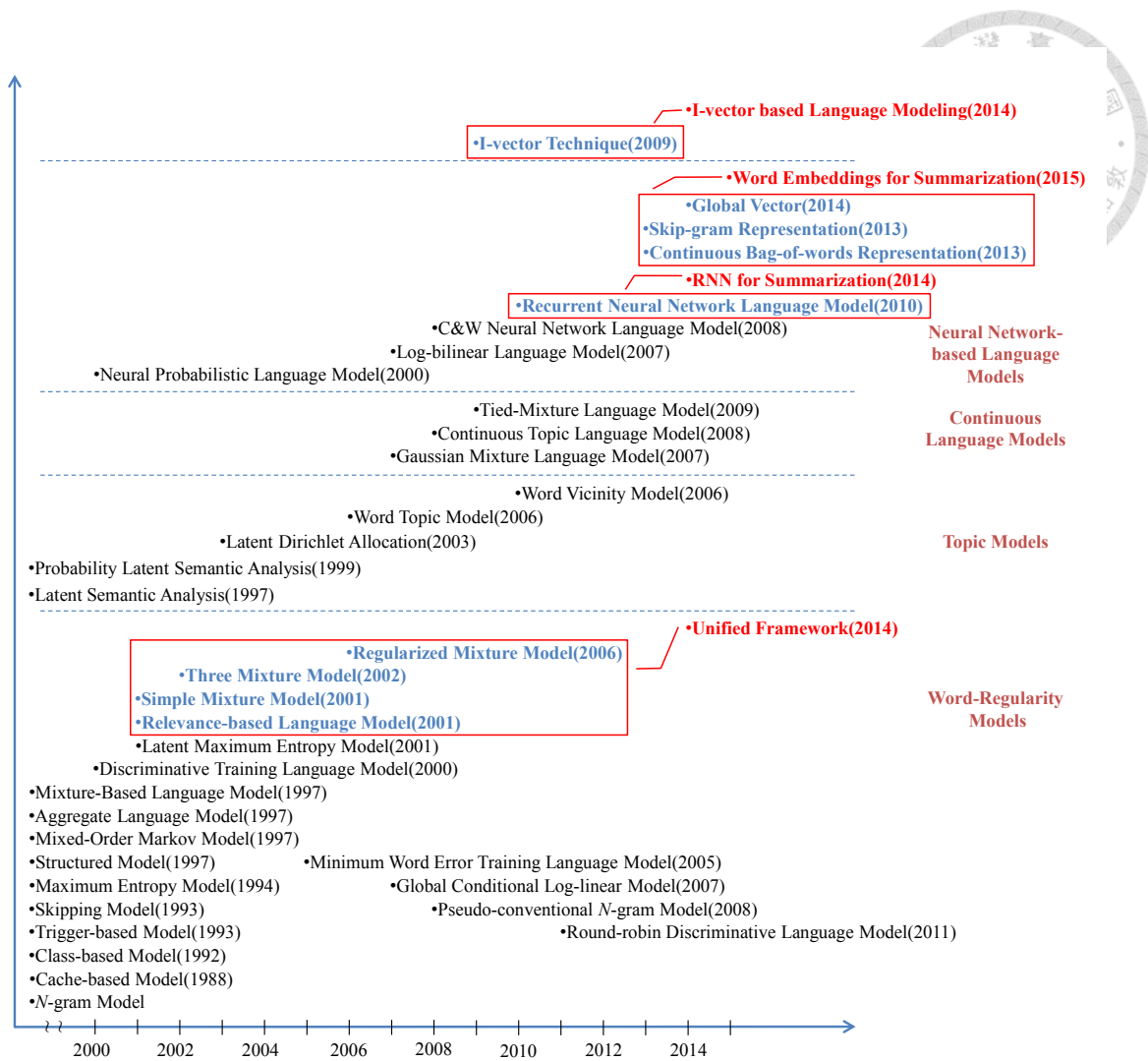


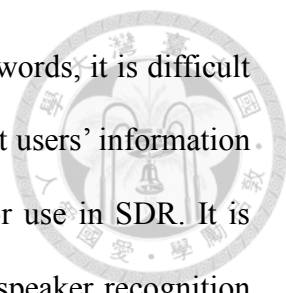
Figure 8.1 The important language models and the proposed frameworks are summarized year by year.

been introduced to extractive summarization.

■ The I-vector based Language Modeling Framework for Retrieval

The i-vector technique, which reduces a series of acoustic feature vectors of a speech utterance to a low-dimensional vector representation, has yielded great performance improvements in language identification and speaker recognition.

In Chapter 5, we adopted this concept for the i-vector based language model (IVLM) for information retrieval. As the major challenge of using IVLM for



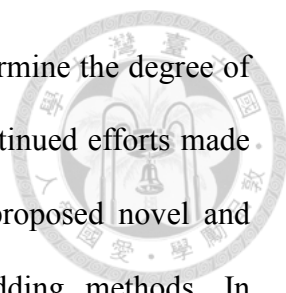
query modeling is that queries usually consist of only a few words, it is difficult to learn reliable representations. To more accurately represent users' information needs, three novel reformulation methods were proposed for use in SDR. It is also expected that conventional language identification and speaker recognition applications can benefit from our methods. In addition, IVLM training also yields a useful by-product: document (or query) and word embeddings.

■ **The RNNLM-based Framework for Summarization**

Language models have been used for unsupervised summarization. However, it remains challenging to formulate the sentence models and to estimate their parameters for each document to be summarized. We proposed a novel recurrent neural network language model using a curriculum learning strategy to render word usage cues and to capture long-span structural information of word co-occurrence relationships within documents in Chapter 6. In addition, we also explored different model complexities and combination strategies, as well as provided in-depth elucidations on the modeling characteristics and the associated summarization performance of various instantiated methods.

■ **The Word Embedding Framework for Summarization**

Recently, word embedding has been a popular research area due to its excellent performance in many natural language processing (NLP)-related tasks. However, as far as we are aware, there has been little work investigating its use in extractive spoken document summarization. The common usage of leveraging word embeddings is to represent the document (or sentence) by averaging the embeddings of the words occurring in the document (or sentence). Then,

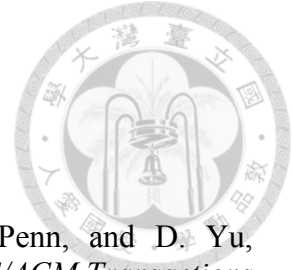


intuitively, the cosine similarity measure can be used to determine the degree of relevance between a pair of representations. Beyond the continued efforts made to improve word representations, in Chapter 7, we have proposed novel and efficient ranking models based on general word embedding methods. In additions, we have also presented a novel probabilistic modeling framework for learning word and sentence representations, which not only inherits the advantages of the original word embedding methods but also boasts a clear and rigorous probabilistic foundation.

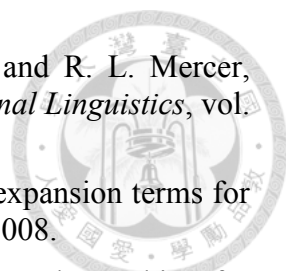
I believe this thesis will help make statistical language modeling more attractive for future research. Still though, there is a need for additional experiments to be conducted and analysis to be made, and there are plenty of related research subtopics that still should be investigated. It is my hope that this work will prove to be a cornerstone for me and others in establishing more elegant, elaborate and powerful methods in the near future.




REFERENCE

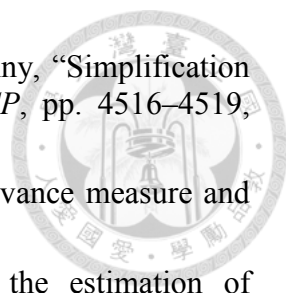


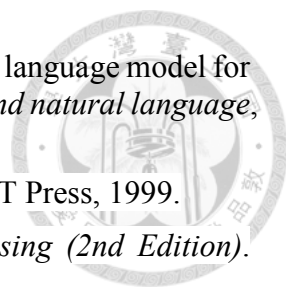
- [1] O. Abdel-Hamid, A. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, “Convolutional neural networks for speech recognition,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 10, pp. 1533–1545, 2014.
- [2] M. Afify, O. Siohan, and R. Sarikaya, “Gaussian mixture language models for speech recognition,” in *Proc. of ICASSP*, pp. IV-29–IV-32, 2007.
- [3] J. Allen, M. S. Hunnicutt, D. H. Klatt, R. C. Armstrong, and D. B. Pisoni, *From Text to Speech: the Mitalk System*. Cambridge University Press, New York, NY, USA, 1987.
- [4] R. Baeza-Yates and B. Ribeiro-Neto, *Modern information retrieval: the concepts and technology behind search*, ACM Press, 2011.
- [5] P. B. Baxendale, “Machine-made index for technical literature-an experiment,” *IBM Journal*, 1958.
- [6] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.
- [7] Y. Bengio, R. Ducharme, and P. Vincent, “A neural probabilistic language model,” in *Proc. of NIPS*, pp. 932–938, 2000.
- [8] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” *Journal of Machine Learning Research* (3), pp. 1137–1155, 2003.
- [9] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, “Curriculum learning,” in *Proc. of ICML*, pp. 41–48, 2009.
- [10] S. Bengio and G. Heigold, “Word embedding for speech recognition,” in *Proc. of Interspeech*, pp. 1053–1057, 2014.
- [11] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2007.
- [12] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent dirichlet allocation,” *Journal of Machine Learning Research*, pp. 993–1022, 2003.
- [13] D. M. Blei and J. D. McAuliffe, “Supervised topic models,” in *Proc. of NIPS*, 2007.
- [14] D. M. Blei and J. Lafferty, “Topic models”, In A. Srivastava and M. Sahami, (eds.), *Text Mining: Theory and Applications*. Taylor and Francis, 2009.
- [15] P. Boersma, “Praat, a system for doing phonetics by computer,” *Glott International*, vol.5, no. 9-10, pp. 341–345, 2001.
- [16] M. Boden, “A guide to recurrent neural networks and backpropagation,” in the Dallas Project, 2002.

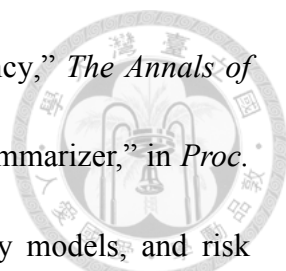
- 
- [17] P. F. Brown, V. J. Della Pietra, P. V. deSouza, J. C. Lai, and R. L. Mercer, “Class-based n-gram models of natural language,” *Computational Linguistics*, vol. 18, no. 4, pp. 467–479, 1992.
- [18] G. Cao, J.-Y. Nie, J. Gao, and S. Robertson, “Selecting good expansion terms for pseudo-relevance feedback,” in *Proc. of SIGIR*, pp. 243–250, 2008.
- [19] J. Carbonell and J. Goldstein, “The use of MMR, diversitybased reranking for reordering documents and producing summaries,” in *Proc. of SIGIR*, pp. 335–336, 1998.
- [20] C. Carpineto and G. Romano, “A survey of automatic query expansion in information retrieval,” *ACM Computing Surveys*, vol. 44, pp.1–56, 2012.
- [21] G. Chechik, V. Sharma, U. Shalit, and S. Bengio, “Large scale online learning of image similarity through ranking,” *Journal of Machine Learning Research (11)*, pp. 1109–1135, 2010.
- [22] C. Chelba, T. J. Hazen, and M. Saraclar, “Retrieval and browsing of spoken content,” *IEEE Signal Processing Magazine*, vol. 25, no. 3, pp. 39–49, 2008.
- [23] B. Chen, H. M. Wang, and L. S. Lee, “A discriminative HMM/n-gram-based retrieval approach for Mandarin spoken documents,” *ACM Transactions on Asian Language Information Processing*, vol. 3, no. 2, pp. 128–145, 2004.
- [24] B. Chen, J.- W. Kuo, and W.-H. Tsai, “Lightly supervised and data-driven approaches to Mandarin broadcast news transcription,” in *Proc. of ICASSP*, 2004.
- [25] B. Chen and Y.-T. Chen, “Extractive spoken document summarization for information retrieval,” *Pattern Recognition Letters*, vol. 29, no. 4, pp. 426–437, 2008.
- [26] B. Chen, “Word topic models for spoken document retrieval and transcription,” *ACM Transactions on Asian Language Information Processing*, vol. 8, no. 1, pp. 2:1–2:27, 2009.
- [27] B. Chen, K.-Y. Chen, P.-N. Chen, and Y.-W. Chen, “Spoken document retrieval with unsupervised query modeling techniques,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no.9, pp. 2602–2612, 2012.
- [28] B. Chen and K. Y. Chen, “Leveraging relevance cues for language modeling in speech recognition,” *Information Processing & Management*, vol. 49, no. 4, pp. 807–816, 2013.
- [29] B. Chen, S. H. Lin, Y. M. Chang, and J. W. Liu, “Extractive speech summarization using evaluation metric-related training criteria,” *Information Processing & Management*, vol. 49, no. 1, pp. 1–12, 2013.
- [30] K. Y. Chen, H. S. Chiu, and B. Chen, “Latent topic modeling of word vicinity information for speech recognition,” in *Proc. of ICASSP*, pp. 5394–5397, 2010.
- [31] K. Y. Chen, H. M. Wang, and B. Chen, “Spoken document retrieval leveraging unsupervised and supervised topic modeling techniques,” *Special Section: Recent Advances in Multimedia Signal Processing Techniques and Applications, IEICE Transactions on Information and Systems*, vol. E95-D, no.5, pp. 1195–1205, 2012.
- [32] K. Y. Chen, H. M. Wang, B. Chen, and H. H. Chen, “Weighted matrix factorization for spoken document retrieval,” in *Proc. of ICASSP*, pp. 8530–8534,

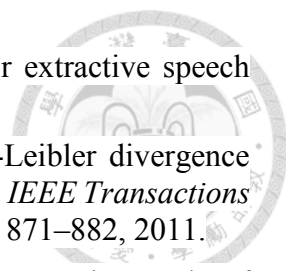
- 2013.
- [33] K. Y. Chen, H. S. Lee, C. H. Lee, H. M. Wang, and H. H. Chen, “A study of language modeling for Chinese spelling check,” in *Proc. of SIGHAN*, pp. 79–83, 2013.
- [34] K. Y. Chen, H. S. Lee, H. M. Wang, B. Chen, and H. H. Chen, “I-vector based language modeling for spoken document retrieval,” in *Proc. of ICASSP*, pp. 7083–7087, 2014.
- [35] K. Y. Chen, S. H. Liu, B. Chen, E. E. Jan, H. M. Wang, W. L. Hsu, and H. H. Chen, “Leveraging effective query modeling techniques for speech recognition and summarization,” in *Proc. of EMNLP*, pp. 1474–1480, 2014.
- [36] K. Y. Chen, S. H. Liu, B. Chen, H. M. Wang, W. L. Hsu, and H. H. Chen, “A recurrent neural network language modeling framework for extractive speech summarization,” in *Proc. of ICME*, pp. 569–574, 2014.
- [37] S. F. Chen and J. Goodman, “An Empirical Study of Smoothing Techniques for Language Modeling,” *Computer Speech and Language*, vol. 13, no. 4, pp. 359–393, 1999.
- [38] Y. T. Chen, B. Chen, and H. M. Wang, “A probabilistic generative framework for extractive broadcast news speech summarization,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 17, no. 1, pp. 95–106, 2009.
- [39] Y. W. Chen, K. Y. Chen, H. M. Wang, and B. Chen, “Effective pseudo-relevance feedback for spoken document retrieval,” in *Proc. of ICASSP*, pp. 8535–8539, 2013.
- [40] Y. Z. Chen, S. H. Wu, C. C. Lu, and T. Ku, “Chinese confusion word set for automatic generation of spelling error detecting template,” in *Proc. of ROCLING*, pp. 359–372, 2009.
- [41] Z. Chen, K. F. Lee, and M. J. Li, “Discriminative training on language model,” in *Proc. of ICSLP*, pp. 493–496, 2000.
- [42] T. K. Chia, K. C. Sim, H. Li, and H. T. Ng, “Statistical lattice-based spoken document retrieval,” *ACM Transactions on Information Systems*, vol. 28, no. 1, pp. 2:1–2:30, 2010.
- [43] S. Clinchant and E. Gaussier, “A theoretical analysis of pseudo-relevance feedback models,” in *Proc. of ICTIR*, pp. 1–6, 2013.
- [44] H. S. Chiu and B. Chen, “Dynamic language model adaptation using word topical mixture models,” in *Proc. of WESPAC*, 2006.
- [45] H. S. Chiu and B. Chen, “Word topical mixture models for dynamic language model adaptation,” in *Proc. of ICASSP*, pp. IV169–IV172, 2007.
- [46] C. H. Chueh and J. T. Chien, “Continuous topic language modeling for speech recognition,” in *Proc. of SLT*, pp. 193–196, 2008.
- [47] M. Collins, “Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms,” in *Proc. of EMNLP*, pp. 1–8, 2002.

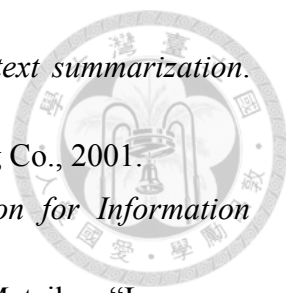
- 
- [48] R. Collobert and J. Weston, “A unified architecture for natural language processing: deep neural networks with multitask learning,” in *Proc. of ICML*, pp. 160–167, 2008.
- [49] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz and Y. Singer, “Online passive-aggressive algorithms,” *Journal of Machine Learning Research* 7, pp. 551–585, 2006.
- [50] B. Croft, D. Metzler and T. Strohman, *Search Engines: Information Retrieval in Practice* (1st ed.). Addison-Wesley Publishing Company, USA, 2009.
- [51] S. Deerwester, S. Dumais, G. Furnas, T. Landauer and R. Harshman, “Indexing by latent semantic analysis”, *Journal of the American Society of Information Science*, vol. 41, no. 6, pp. 391–407, 1990.
- [52] A. P. Dempster, N. M. Laird and D. B. Rubin, “Maximum likelihood from incomplete data via the EM algorithm,” *Journal of Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.
- [53] L. F. D’Haro, O. Glembek, O. Plchot, P. Matejka, M. Soufifar, R. Cordoba, and J. Cernocky, “Phonotactic language recognition using i-vectors and phoneme posterigram counts,” in *Proc. of Interspeech*, pp. 42–45, 2012.
- [54] J. V. Dillon and K. Collins-Thompson, “A unified optimization framework for robust pseudo-relevance feedback algorithms,” in *Proc. of CIKM*, pp. 1069–1078, 2010.
- [55] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization,” *Journal of Machine Learning Research* (12), pp. 2121–2159, 2011.
- [56] J. L. Elman, “Finding structure in time,” *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.
- [57] J. L. Elman, “Learning and development in neural networks: the importance of starting small,” *Cognition*, vol. 48, pp. 71–99, 1993.
- [58] G. Erkan and D. R. Radev, “LexRank: Graph-based lexical centrality as salience in text summarization”, *Journal of Artificial Intelligent Research*, vol. 22, no. 1, pp. 457–479, 2004.
- [59] S. Furui, T. Kikuchi, Y. Shinnaka, and C. Hori, “Speech-to-text and speech-to-speech summarization of spontaneous speech,” *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 4, pp. 401–408, 2004.
- [60] S. Furui, L. Deng, M. Gales, H. Ney, and K. Tokuda, “Fundamental technologies in modern speech recognition,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 16–17, 2012.
- [61] M. Galley, “Skip-chain conditional random field for ranking meeting utterances by importance,” in *Proc. of EMNLP*, pp. 364–372, 2006.
- [62] D. Garcia-Romero and C. Y. Espy-Wilson, “Analysis of i-vector length normalization in speaker recognition systems,” in *Proc. of Interspeech*, pp. 249–252, 2011.
- [63] J. Garofolo, G. Auzanne, and E. Voorhees, “The TREC spoken document retrieval track: A success story,” in *Proc. TREC*, pp. 107–129, 2000.

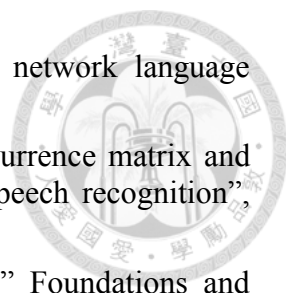
- 
- [64] O. Glembek, L. Burget, P. Matejka, M. Karafiat, and P. Kenny, “Simplification and optimization of i-vector extraction,” in *Proc. of ICASSP*, pp. 4516–4519, 2011.
- [65] Y. Gong and X. Liu, “Generic text summarization using relevance measure and latent semantic analysis,” in *Proc. of SIGIR*, pp. 19–25, 2001.
- [66] I. J. Good, “The population frequencies of species and the estimation of population parameters,” *Biometrika*, 40:16–264, 1953.
- [67] T. L. Griffiths and M. Steyvers, “Finding scientific topics,” in *Proc. of PNAS*, pp. 5228–5235, 2004.
- [68] M. T. Hagan and M. B. Menhaj, “Training feedforward networks with the Marquardt algorithm,” *IEEE Transactions on Neural Networks*, vol.5, no.6, pp. 989–993, 1994.
- [69] A. Haghighi and L. Vanderwende, “Exploring content models for multi-document summarization,” in *Proc. of HLT/NAACL*, pp. 362–370, 2009.
- [70] D. F. Harwath, T. J. Hazen, and J. R. Glass, “Zero resource spoken audio corpus analysis,” in *Proc. of ICASSP*, pp. 8555–8559, 2013.
- [71] T. Hasan, R. Saeidi, J. H. L. Hansen, and D. A. van Leeuwen, “Duration mismatch compensation for i-vector based speaker recognition systems,” in *Proc. of ICASSP*, pp. 7663–7667, 2013.
- [72] V. Hautamaki, Y. C. Cheng, P. Rajan, and C. H. Lee, “Minimax i-vector extractor for short duration speaker verification,” in *Proc. of Interspeech*, pp. 3708–3712, 2013.
- [73] G. Heigold, H. Ney, R. Schluter, and S. Wiesler, “Discriminative training for automatic speech recognition: Modeling, criteria, optimization, implementation, and performance,” *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 58–69, 2012.
- [74] D. Hiemstra, S. Robertson, and H. Zaragoza, “Parsimonious language models for information retrieval,” in *Proc. of SIGIR*, pp. 178–185, 2004.
- [75] T. Hofmann, “Probabilistic latent semantic indexing,” in *Proc. of SIGIR*, pp. 50–57, 1999.
- [76] T. Hofmann, “Unsupervised learning by probabilistic latent semantic analysis,” *Machine Learning*, vol. 42, pp. 177–196, 2001.
- [77] C. L. Huang, B. Ma, H. Li, and C. H. Wu, “Speech indexing using semantic context inference,” in *Proc. of Interspeech*, pp. 717–720, 2011.
- [78] S. F. Huang and S. Renals, “Hierarchical Pitman-Yor language models for ASR in meetings,” in *Proc. of ASRU*, pp. 124–129, 2007.
- [79] X. Huang, A. Acero, H.-W. Hon, “*Spoken Language Processing: A Guide to Theory, Algorithm, and System Development* (1st ed.)” Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.
- [80] H. Jaeger, “A tutorial on training recurrent neural networks, covering BPPT, RTRL, EKF and the echo state network approach,” *GMD Report 159, German National Research Center for Information Technology*, 2002.

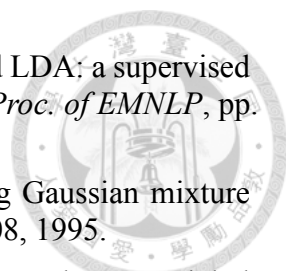
- 
- [81] F. Jelinek, B. Meriello, S. Roukos, and M. Strauss, “A dynamic language model for speech recognition,” in *Proc. of DARPA workshop on speech and natural language*, pp. 293–295, 1991.
- [82] F. Jelinek, *Statistical Methods for Speech Recognition*. The MIT Press, 1999.
- [83] D. Jurafsky and J. H. Martin, *Speech and Language Processing (2nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2009.
- [84] M. Kageback, O. Mogren, N. Tahmasebi, and D. Dubhashi, “Extractive summarization using continuous vector space models,” in *Proc. of CVSC*, pp. 31–39, 2014.
- [85] A. Kanagasundaram, R. Vogt, D. Dean, S. Sridharan, and M. Mason, “I-vector based speaker recognition on short utterances,” in *Proc. of Interspeech*, pp. 2341–2344, 2011.
- [86] A. Kanagasundaram, D. Dean, J. Gonzalez-Dominguez, S. Sridharan, D. Ramos, and J. Gonzalez-Rodriguez, “Improving short utterance based i-vector speaker recognition using source and utterance-duration normalization techniques,” in *Proc. of Interspeech*, pp. 2465–2469, 2013.
- [87] A. Karpathy and F. F. Li, “Deep visual-semantic alignments for generating image descriptions,” *arXiv:1412.2306*, 2014.
- [88] S. M. Katz, “Estimation of probabilities from sparse data for the language model component of a speech recognizer,” *IEEE Transactions on Audio, Speech, and Signal Processing*, vol. 35, no. 3, pp. 400–401, 1987.
- [89] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, “Joint factor analysis versus eigenchannels in speaker recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1435–1447, 2007.
- [90] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, “Speaker and session variability in GMM-based speaker verification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 4, pp. 1448–1460, 2007.
- [91] P. Kenny, T. Stafylakis, P. Ouellet, Md. J. Alam, and P. Dumouchel, “PLDA for speaker verification with utterances of arbitrary duration,” in *Proc. of ICASSP*, pp. 7649–7653, 2013.
- [92] R. Kneser and H. Ney, “Improved backing-off for m-gram language modeling,” in *Proc. of ICASSP*, pp. 181–184, 1995.
- [93] S. Kombrink, T. Mikolov, M. Karafiat, and L. Burget, “Recurrent neural network based language modeling in meeting recognition,” in *Proc. of Interspeech*, pp. 2877–2880, 2011.
- [94] A. Krizhevsky, I. Sutskever and G. E. Hinton, “ImageNet classification with deep convolutional neural networks,” in *Proc. of NIPS*, pp. 1–9, 2012.
- [95] R. Kuhn, “Speech recognition and the frequency of recently used words: A modified Markov model for natural language,” in *Proc. of COLING*, pp. 348–350, 1988.
- [96] R. Kuhn and R. D. Mori, “A cache-based natural language model for speech recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 6, pp. 570–583, 1990.

- 
- [97] S. Kullback and R. A. Leibler, “On information and sufficiency,” *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.
- [98] J. Kupiec, J. Pedersen, and F. Chen, “A trainable document summarizer,” in *Proc. of SIGIR*, pp. 68–73, 1995.
- [99] J. Lafferty and C. Zhai, “Document language models, query models, and risk minimization for information retrieval,” in *Proc. of SIGIR*, pp. 111–119, 2001.
- [100] M. Larson and G. J. F. Jones, “Spoken content retrieval: A survey of techniques and technologies,” *Foundations and Trends in Information Retrieval*, vol. 5, no. 4–5, pp. 235–422, 2012.
- [101] R. Lau, R. Rosenfeld and S. Roukos, “Trigger-based language models: a maximum entropy approach,” in *Proc. of ICASSP*, pp. II45–II48, 1993.
- [102] V. Lavrenko and B. Croft, “Relevance-based language models,” in *Proc. of SIGIR*, pp. 120–127, 2001.
- [103] V. Lavrenko, *A Generative Theory of Relevance*. PhD thesis, University of Massachusetts, Amherst, 2004.
- [104] LDC, “Project topic detection and tracking,” *Linguistic Data Consortium*, 2000.
- [105] H. S. Le, I. Oparin, A. Messaoudi, A. Allauzen, J. L. Gauvain, and F. Yvon, “Large vocabulary SOUL neural network language models,” in *Proc. of Interspeech*, pp. 1469–1472, 2011.
- [106] H. Y. Lee and L. S. Lee, “Improved semantic retrieval of spoken content by document/query expansion with random walk over acoustic similarity graphs,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 1, pp. 80–94, 2014.
- [107] K. S. Lee, W. B. Croft, and J. Allan, “A cluster-based resampling method for pseudo-relevance feedback,” in *Proc. of SIGIR*, pp. 235–242, 2008.
- [108] K. S. Lee and W. B. Croft, “A deterministic resampling method using overlapping document clusters for pseudo-relevance feedback,” *Inf. Process. Manage.*, vol. 49, no. 4, pp. 792–806, 2013.
- [109] L. S. Lee and B. Chen, “Spoken document understanding and organization,” *IEEE Signal Processing Magazine*, vol. 22, no. 5, pp. 42–60, 2005.
- [110] C. J. Leggetter and P. C. Woodland, “Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models,” *Computer Speech and Language*, vol. 9, pp. 171–185, 1995.
- [111] O. Levy and Y. Goldberg, “Neural word embedding as implicit matrix factorization,” in *Proc. of NIPS*, pp. 2177–2185, 2014.
- [112] D. Li and D. Yu, *Deep Learning: Methods and Applications*. Foundations and Trends in Signal Processing, Now Publishers, June 2014.
- [113] C. Y. Lin. 2003. ROUGE: Recall-oriented Understudy for Gisting Evaluation. Available: <http://haydn.isi.edu/ROUGE/>.
- [114] H. Lin and J. Bilmes, “Multi-document summarization via budgeted maximization of submodular functions,” in *Proc. of NAACL HLT*, pp. 912–920, 2010.

- 
- [115] S. H. Lin and B. Chen, “A risk minimization framework for extractive speech summarization,” in *Proc. of ACL*, pp. 79–87, 2010.
- [116] S. H. Lin, Y. M. Yeh, and B. Chen, “Leveraging Kullback-Leibler divergence measures and information-rich cues for speech summarization,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 871–882, 2011.
- [117] S. H. Lin, Y. T. Chen, H. M. Wang, and B. Chen, “A comparative study of probabilistic ranking models for Chinese spoken document summarization,” *ACM Transactions on Asian Language Information Processing*, vol. 8, no. 1, pp. 3:1–3:23, 2009.
- [118] C. L. Liu, M. H. Lai, K. W. Tien, Y. H. Chuang, S. H. Wu, and C. Y. Lee, “Visually and phonologically similar characters in incorrect chinese words: analyses, identification, and applications,” *ACM Transactions on Asian Language Information Processing*, vol. 10, no. 2, pp. 1–39, 2011.
- [119] F. Liu and Y. Liu, “Unsupervised language model adaptation incorporating named entity information,” in *Proc. of ACL*, pp. 672–769, 2007.
- [120] S. H. Liu, K. Y. Chen, Y. L. Hsieh, B. Chen, H. M. Wang, H. C. Yen and W. L. Hsu, “Effective pseudo-relevance feedback for language modeling in extractive speech summarization,” in *Proc. of ICASSP*, pp. 3226–3230, 2014.
- [121] S. H. Liu, K. Y. Chen, Y. L. Hsieh, B. Chen, H. M. Wang, H. C. Yen and W. L. Hsu, “Enhanced language modeling for extractive speech summarization with sentence relatedness information,” in *Proc. of Interspeech*, pp. 1865–1869, 2014.
- [122] S. H. Liu, K. Y. Chen, B. Chen, E. E. Jan, H. M. Wang, H. C. Yen and W. L. Hsu, “A margin-based discriminative modeling approach for extractive speech summarization,” in *Proc. of APSIPA*, pp. 1–6, 2014.
- [123] X. Liu, Y. Wang, X. Chen, M. Gales, and P. Woodland, “Efficient lattice rescoring using recurrent neural network language models,” in *Proc. of ICASSP*, pp. 4941–4945, 2014.
- [124] X. Liu, J. Gao, X. He, L. Deng, K. Duh and Y. Y. Wang, “Representation learning using multi-task deep neural networks for semantic classification and information retrieval,” in *Proc. of NAACL*, 2015.
- [125] Y. Liu and D. Hakkani-Tur, “Speech Summarization,” *Chapter 13 in Spoken Language Understanding: Systems for Extracting Semantic Information from Speech*, G. Tur and R. D. Mori (Eds), Wiley, New York, 2011.
- [126] Y. Lu, Q. Mei and C. X. Zhai, “Investigating task performance of probabilistic topic models: an empirical study of PLSA and LDA,” *Information Retrieval*, vol.14, no. 2, pp. 178–203, 2011.
- [127] Y. Lv and C. Zhai, “A comparative study of methods for estimating query language models with pseudo feedback,” in *Proc. of CIKM*, pp. 1895–1898, 2009.
- [128] Y. Lv and C. Zhai, “Positional relevance model for pseudo-relevance feedback,” in *Proc. of SIGIR*, pp. 579–586, 2010.
- [129] A. L. Maas and A. Y. Ng, “A probabilistic model for semantic word vectors,” in *Proc. of NIPS Workshop*, 2010.

- 
- [130] I. Mani and M. T. Maybury (Eds.). *Advances in automatic text summarization*. Cambridge, MA: MIT Press, 1999.
- [131] I. Mani, *Automatic Summarization*. John Benjamins Publishing Co., 2001.
- [132] C. D. Manning, P. Raghavan, and H. Schutze, *Introduction for Information Retrieval*, Cambridge University Press, 2008.
- [133] D. Martinez, O. Plchot, L. Burget, O. Glembek, and P. Matejka, “Language recognition in ivector space,” in *Proc. of Interspeech*, pp. 861–864, 2011.
- [134] W. McCulloch and W. Pitts, “A logical calculus of ideas immanent in nervous activity,” *Bulletin of Mathematical Biophysics* 5 (4), pp. 115–133, 1943.
- [135] R. McDonald, “A study of global inference algorithms in multi-document summarization,” in *Proc. of ECIR*, pp. 557–564, 2007.
- [136] K. McKeown, J. Hirschberg, M. Galley, and S. Maskey, “From text to speech summarization,” in *Proc. of ICASSP*, pp. 997–1000, 2005.
- [137] G. Miao, Z. Guan, L. E. Moser, X. Yan, S. Tao, N. Anerousis and J. Sun, “Latent association analysis of document pairs,” in *Proc. of KDD*, pp. 1415–1423, 2012.
- [138] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, and S. Khudanpur, “Recurrent neural network based language model,” in *Proc. of Interspeech*, pp. 1045–1048, 2010.
- [139] T. Mikolov, S. Kombrink, L. Burget, J. Cernocky, and S. Khudanpur, “Extensions of recurrent neural network language model,” in *Proc. of ICASSP*, pp. 5528–5531, 2011.
- [140] T. Mikolov and G. Zweig, “Context dependent recurrent neural network language model,” in *Proc. of SLT*, pp. 234–239, 2012.
- [141] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient estimation of word representations in vector space,” in *Proc. of ICLR*, pp. 1–12, 2013.
- [142] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, “Distributed representations of words and phrases and their compositionality,” in *Proc. of ICLR*, pp. 1–9, 2013.
- [143] D. R. H. Miller, T. Leek, and R. M. Schwartz, “A hidden Markov model information retrieval system,” in *Proc. of SIGIR*, pp. 214–221, 1999.
- [144] G. Miller and W. Charles, “Contextual correlates of semantic similarity,” *Language and Cognitive Processes*, vol. 6, no. 1, pp. 1–28, 1991.
- [145] T. Minka and J. Lafferty, “Expectation-propagation for the generative aspect model,” in *Proc. of UAI*, pp. 352–359, 2002.
- [146] A. Mnih and G. Hinton, “Three new graphical models for statistical language modeling,” in *Proc. of ICML*, pp. 641–648, 2007.
- [147] A. Mnih and K. Kavukcuoglu, “Learning word embeddings efficiently with noise-contrastive estimation,” in *Proc. of NIPS*, pp. 2265–2273, 2013.
- [148] A. Mohamed, G.E. Dahl, and G. Hinton, “Acoustic modeling using deep belief networks,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

- 
- [149] F. Morin and Y. Bengio, “Hierarchical probabilistic neural network language model,” in *Proc. of AISTATS*, pp. 246–252, 2005.
- [150] W. Naptali, M. Tsuchiya, and S. Nakagawa, “Word co-occurrence matrix and context dependent class in LSA based language model for speech recognition”, *International Journal of Computers*, pp. 85–95, 2009.
- [151] A. Nenkova and K. McKeown, “Automatic summarization,” *Foundations and trends in information retrieval*, vol. 5, no. 2–3, pp. 103–233, 2011.
- [152] M. Norouzi, D. J. Fleet, and R. Salakhutdinov, “Hamming distance metric learning,” in *Proc. of NIPS*, pp. 1070–1078, 2012.
- [153] T. Oba, T. Hori and A. Nakamura, “A comparative study on methods of weighted language model training for reranking LVCSR n-best hypotheses,” in *Proc. of ICASSP*, pp. 5126–5129, 2010.
- [154] T. Oba, T. Hori, A. Nakamura and A. Ito, “Round-robin duel discriminative language models,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 4, pp. 1244–1255, 2012.
- [155] F. J. Och, “Minimum error rate training in statistical machine translation,” in *Proc. of ACL*, pp. 160–167, 2003.
- [156] S. Ortman, H. Ney, and X. Aubert, “A word graph algorithm for large vocabulary continuous speech recognition,” *Computer Speech and Language*, pp. 43–72, 1997.
- [157] D. O’Shaughnessy, L. Deng, and H. Li, “Speech information processing: Theory and applications,” *Proceedings of the IEEE*, vol. 101, no. 5, pp. 1034–1037, 2013.
- [158] M. Ostendorf, “Speech technology and information access,” *IEEE Signal Processing Magazine*, vol. 25, no. 3, pp. 150–152, 2008.
- [159] H. Palangi, L. Deng, Y. Shen, J. Gao, X. He, J. Chen, X. Song, and R. Ward, “Deep sentence embedding using the long short term memory network: analysis and application to information retrieval,” in *arXiv:1502.06922*, 2015.
- [160] S. Pang, and N. Kasabov, “Inductive vs transductive inference, global vs local models: SVM, TSVM, and SVMT for gene expression classification problems,” in *Proc. of IJCNN*, pp. 1197–1202, 2004.
- [161] S. Parlak and M. Saraçlar, “Performance analysis and improvement of Turkish broadcast news retrieval,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 3, pp. 731–743, 2012.
- [162] R. Pascanu *et al.*, “On the difficulty of training recurrent neural networks,” in *Proc. JMLR: W&CP*, pp. 1310–1318, 2013.
- [163] G. Penn and X. Zhu, “A critical reassessment of evaluation baselines for speech summarization,” in *Proc. of ACL*, pp. 470–478, 2008.
- [164] J. Pennington, R. Socher, and C. D. Manning, “GloVe: Global vector for word representation,” in *Proc. of EMNLP*, pp. 1532–1543, 2014.
- [165] J. M. Ponte and W. B. Croft, “A language modeling approach to information retrieval,” in *Proc. of SIGIR*, pp. 275–281, 1998.
- [166] L. Qiu, Y. Cao, Z. Nie, and Y. Rui, “Learning word representation considering proximity and ambiguity,” in *Proc. of AAAI*, pp. 1572–1578, 2014.

- 
- [167] D. Ramage, D. Hall, R. Nallapati, and C. D. Manning, “Labeled LDA: a supervised topic model for credit attribution in multi-labeled corpora,” in *Proc. of EMNLP*, pp. 248–256, 2009.
- [168] D. A. Reynolds, “Speaker identification and verification using Gaussian mixture speaker models,” *Speech Commun.*, vol. 17, no. 1–2, pp. 91–108, 1995.
- [169] K. Riedhammer, B. Favre, and D. Hakkani-Tur, “Long story short – Global unsupervised models for keyphrase based meeting summarization,” *Speech Communication*, vol. 52, no. 10, pp. 801–815, 2010.
- [170] B. Roark, M. Saraclar, and M. Collins, “Corrective language modeling for large vocabulary ASR with the perceptron algorithm,” in *Proc. of ICASSP*, pp. 749–752, 2004.
- [171] B. Roark, M. Saraclar, and M. Collins, “Discriminative n-gram language modeling,” *Computer Speech & Language*, vol. 21, no. 2, pp. 373–392, 2007.
- [172] S. E. Robertson, “On term selection for query expansion,” *Journal of Documentation*, vol. 46, no. 4, pp. 359–364, 1990.
- [173] J. Rocchio, “Relevance feedback in information retrieval,” in G. Salton (Ed.), *The SMART Retrieval System: Experiments in Automatic Document Processing*, Prentice Hall, pp. 313–323, 1971.
- [174] R. Rosenfeld, “A maximum entropy approach to adaptive statistical language modeling,” *Computer, Speech, and Language*, vol. 10, pp. 187–228, 1996.
- [175] R. Rosenfeld, “A whole sentence maximum entropy language model,” in *Proc. of ASRU*, pp. 230–237, 1997.
- [176] R. Rosenfeld, “Two decades of statistical language modeling: where do we go from here,” in *Proc. of IEEE*, vol. 88, no. 8, pp. 1270–1278, 2000.
- [177] G. Salton, *Automatic Information Organization and Retrieval*. New York: McGraw-Hill, 1968.
- [178] G. Salton, A. Wong, and C. S. Yang, “A vector space model for automatic indexing,” *Communications of the ACM*, vol. 18, no. 11, pp. 613–620, 1975.
- [179] G. Salton and M. J. McGill, *Introduction to Modern Information Retrieval*. New York: McGraw-Hill, 1983.
- [180] R. Sarikaya, M. Afify, and B. Kingsbury, “Tied-mixture language modeling in continuous space,” in *Proc. of NAACL*, pp. 459–467, 2009.
- [181] A. K. Sarker, D. Matrouf, P. M. Bousquet, and J. F. Bonastre, “Study of the effect of i-vector modeling on short and mismatch utterance duration for speaker verification,” in *Proc. of Interspeech*, pp. 2662–2665, 2012.
- [182] X. Shen and C. Zhai, “Active feedback in ad hoc information retrieval,” in *Proc. of SIGIR*, pp. 55–66, 2005.
- [183] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng, “Semantic compositionality through recursive matrix-vector spaces,” in *Proc. of EMNLP*, pp. 1201–1211, 2012.
- [184] M. Soufifar, S. Cumani, L. Burget, and J. Cernocky, “Discriminative classifiers for phonotactic language recognition with ivectors,” in *Proc. of ICASSP*, pp.

4853–4856, 2012.

- [185] M. Soufifar, M. Kockmann, L. Burget, and O. Plchot, O. Glembek, and T. Svendsen, “I-vector approach to phonotactic language recognition,” in *Proc. of Interspeech*, pp. 2913–2916, 2011.
- [186] F. Song and W. B. Croft, “A general language model for information retrieval,” in *Proc. of CIKM*, pp. 316–321, 1999.
- [187] A. Stolcke, “SRILM—An extensible language modeling toolkit,” in *Proc. of Interspeech*, pp. 901–904, 2005.
- [188] Y.-C. Tam and T. Schultz, “Dynamic language model adaptation using variational Bayes inference,” in *Proc. of Interspeech*, pp. 5–8, 2005.
- [189] D. Tang, F. Wei, N. Yang, M. Zhou, T. Liu, and B. Qin “Learning sentiment-specific word embedding for twitter sentiment classification” in *Proc. of ACL*, pp. 1555–1565, 2014.
- [190] T. Tao and C. Zhai, “Regularized estimation of mixture models for robust pseudo-relevance feedback,” in *Proc. of SIGIR*, pp. 162–169, 2006.
- [191] V. T. Turunen and M. Kurimo, “Indexing confusion networks for morph-based spoken document retrieval,” in *Proc. of SIGIR*, pp. 631–638, 2007.
- [192] X. Wan and J. Yang, “Multi-document summarization using cluster-based link analysis,” in *Proc. of SIGIR*, pp. 299–306, 2008.
- [193] H.-M. Wang, B. Chen, J.-W. Kuo, and S.-S. Cheng, “MATBN: A Mandarin Chinese broadcast news corpus,” *International Journal of Computational Linguistics & Chinese Language Processing*, vol. 10, no. 2, pp. 219–236, 2005.
- [194] X. Wang, H. Fang, and C. Zhai, “A study of methods for negative relevance feedback,” in *Proc. of SIGIR*, pp. 219–226, 2008.
- [195] X. Wei and W. B. Croft, “LDA-based document models for ad-hoc retrieval,” in *Proc. of SIGIR*, pp. 178–185, 2006.
- [196] E. W D. Whittaker and P. C. Woodland, “Efficient class-based language modelling for very large vocabularies,” in *Proc. of ICASSP*, pp. 545–548, 2001.
- [197] S.-H. Wu, Y.-Z. Chen, P.-C. Yang, T. Ku, and C.-L. Liu, “Reducing the false alarm rate of Chinese character error detection and correction,” in *Proc. of SIGHAN*, 2010.
- [198] S. Xie and Y. Liu, “Improving supervised learning for meeting summarization using sampling and regression,” *Computer Speech & Language*, vol. 24, no. 3, pp. 495–514, 2010.
- [199] Z. Xu, R. Akella, and Y. Zhang, “Incorporating diversity and density in active learning for relevance feedback,” in *Proc. of ECIR*, pp. 245–257, 2007.
- [200] S. Young, D. Kershaw, J. Odell, D. Ollason, V. Valtchev, and P. Woodland, *The HTK book version 3.4*. Cambridge University Press, 2006.
- [201] X. Yi and J. Allan, “A comparative study of utilizing topic models for information retrieval,” in *Proc. of ECIR*, pp. 29–41, 2009.
- [202] C. Zhai and J. Lafferty, “A study of smoothing methods for language models

- applied to ad hoc information retrieval,” in *Proc. of SIGIR*, pp. 334–342, 2001.
- [203] C. Zhai and J. Lafferty, “Model-based feedback in the language modeling approach to information retrieval,” in *Proc. of CIKM*, pp. 403–410, 2001.
- [204] C. Zhai, “Statistical language models for information retrieval: a critical review,” *Foundations and Trends in Information Retrieval*, vol. 2, no. 3, pp. 137–213, 2008.
- [205] J. Zhang and P. Fung, “Speech summarization without lexical features for Mandarin broadcast news”, in *Proc. of NAACL HLT, Companion Volume*, pp. 213–216, 2007.
- [206] J. J. Zhang, R. H. Y. Chan, and P. Fung, “Extractive speech summarization using shallow rhetorical structure modeling,” *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 6, pp. 1147–1157, 2010.
- [207] Y. Zhang, J. Callan, and T. Minka, “Novelty and redundancy detection in adaptive filtering,” in *Proc. of SIGIR*, pp. 81–88, 2002.
- [208] Z. Zhou, J. Gao, F. K. Soong, and H. Meng, “A comparative study of discriminative methods for reranking LVCSR n-best hypotheses in domain adaptation and generalization,” in *Proc. of ICASSP*, pp. 141–144, 2006.